

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Стохастичне моделювання кінцевих вузлів  
транспортних потоків

Виконав: студент IV курсу, групи СНЗс-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Гонтар В.О.  
(підпис) (прізвище та ініціали)

Керівник Кунанець Н.Е.  
(підпис) (прізвище та ініціали)

Нормоконтроль Шимчук Г.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.  
(підпис) (прізвище та ініціали)

Рецензент Тиш Є.В.  
(підпис) (прізвище та ініціали)

Тернопіль - 2022

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

«\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Гонтару Вадиму Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Стохастичне моделювання кінцевих вузлів транспортних потоків

Керівник роботи Кунанець Наталія Едуардівна, д. н. соц. ком., проф. каф. КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «23» 03 2022 року № 4/7-172

2. Термін подання студентом завершеної роботи 12.06.2022р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

1. Аналітична частина. 1.1. Опис предметної області. 1.2. КВ. 1.3. ТП, що рахується в КВ.

1.4. Опис моделі, що розробляється. 2. Теоретико-практична частина.

2.1. Введення в теорію СМО. 2.2. Практична реалізація імітаційної моделі

3. Результати моделювання. 3.1. Параметри модельованого КВ.

3.2. Виведення результатів у вигляді тексту. 3.3 Виведення результатів у вигляді графіків.

4. Безпека життєдіяльності, основи хорони праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титулка. 2. Актуальність. 3. Мета, задачі дослідження. 4, 5. Опис предметної області

6. Повна схема моделі КВ. 7. Фреймворк Salabim. 8. Код класу, що описує поведінку ТЗ на парковці. 9. Блок-схеми поведінки водія на різних фазах.

10. Трасе -лог моделювання, Гістограма зайнятих місць на стоянці впродовж одного дня.

11. Графіки із результатами моделювання. 12. Основні результати проведеного дослідження

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання \_\_\_\_\_ 2022 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	23.03 – 25.03	<i>Виконано</i>
2.	Підбір джерел про стохастичне моделювання кінцевих вузлів транспортних потоків	26.03 – 15.04	<i>Виконано</i>
3.	Опрацювання джерел про стохастичне моделювання кінцевих вузлів транспортних потоків	16.04 – 28.04	<i>Виконано</i>
4.	Виконання дослідження щодо стохастичного моделювання кінцевих вузлів транспортних потоків	29.04 – 10.05	<i>Виконано</i>
5	Розроблення програмного коду	11.05 – 15.05	<i>Виконано</i>
6.	Оформлення розділу «Аналітична частина»	15.05 – 20.05	<i>Виконано</i>
7.	Оформлення розділу «Теоретико практична частина»	21.05 – 26.05	<i>Виконано</i>
8.	Оформлення розділу «Результати моделювання»	27.05 – 31.05	<i>Виконано</i>
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	12.05 – 16.05	<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи	16.05 – 26.05	<i>Виконано</i>
11.	Нормоконтроль	27.05 – 02.06	<i>Виконано</i>
12.	Перевірка на плагіат	03.06 – 07.06	<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи	08.06 – 10.06	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	13.06	

Студент

\_\_\_\_\_  
(підпис)

Гонтар В.О.

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Кунанець Н.Е.

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

Стохастичне моделювання кінцевих вузлів транспортних потоків // Гонтар Вадим Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СНзс-42 // Тернопіль, 2022 // С. – 53, рис. – 24, табл. – 1, слайдів – 12, бібліогр. – 17 .

Ключові слова: ТРАНСПОРТНИЙ ПОТІК, ПАРКУВАННЯ, СТОХАСТИЧНЕ МОДЕЛЮВАННЯ, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, PYTHON, SALABIM

Кваліфікаційна робота присвячена стохастичному моделюванню кінцевих вузлів транспортних потоків. Докладно розглянуто поняття транспортної мережі, кінцевого вузла. Проаналізовано підходи до проектування парковок.

Розроблена стохастична модель транспортного потоку на парковці. Розроблено дослідницький прототип програми, що дозволяє проводити імітаційне моделювання кінцевих вузлів. Проведено імітаційне моделювання на розробленій моделі. Модель представлена як багатофазна система масового обслуговування з вхідним нестационарним пуасонівським потоком, а сам процес моделювання виконується імітаційно із застосуванням розробленого додатку у середовищі Salabim.

Результати моделювання представляються в текстовому та графічному вигляді виражають ефективність паркування при визначених параметрах.

## ANNOTATION

Stochastic modeling of terminal nodes of transport flows // Hontar Vadim // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2022 // P. - 53, Fig. - 24, Table - 1, Slide - 12, References - 17.

Keywords: TRANSPORT FLOW, PARKING, STOCHASTIC MODELING, SIMULATION MODELING, PYTHON, SALABIM

Thesis deals with the stochastic modeling of end nodes of traffic flows. The concept of transport network, end node is considered in detail. Approaches to parking design are analyzed.

A stochastic model of traffic flow in the parking lot has been developed. A research prototype of the program has been developed, which allows to carry out simulation modeling of end nodes. Simulation modeling on the developed model is carried out. The model is presented as a multiphase queuing system with incoming non-stationary Poisson flow, and the simulation process itself is performed simulated using the developed application in the Salabim environment.

The simulation results are presented in text and graphical form expressing the efficiency of parking at certain parameters.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ**

СМО – система масового обслуговування

ТЗ – транспортний засіб

ТП – транспортний потік

КВ – кінцевий вузол

Авт – автомобіль

## ЗМІСТ

Вступ.....	7
1 Аналітична частина.....	8
1.1 Опис предметної області.....	8
1.2 КВ.....	10
1.3 ТП, що рухається в КВ.....	12
1.4 Опис моделі, що розробляється.....	13
1.4.1 Модель вхідного потоку.....	13
1.4.2 Модель КВ.....	15
2 Теоретико-практична частина.....	16
2.1 Введення в теорію СМО.....	16
2.1.1 СМО для моделювання в'їзду на КВ.....	17
2.1.2 СМО для моделювання стоянки на КВ.....	18
2.1.3 СМО для моделювання виїзду з КВ.....	19
2.1.4 Час обслуговування.....	20
2.2 Практична реалізація імітаційної моделі.....	20
2.2.1 Фреймворк Salabim.....	20
2.2.2 Основні компоненти програми.....	23
3 Результати моделювання.....	36
3.1 Параметри модельованого КВ.....	36
3.2 Виведення результатів у вигляді тексту.....	37
3.3 Виведення результатів у вигляді графіків.....	40
4 Безпека життєдіяльності, основи охорони праці.....	45
4.1 Вимоги ергономіки до організації робочого місця оператора ПК.....	45
4.2 Заходи захисту від випромінювань оптичного діапазону.....	47
Висновки.....	50
Перелік використаних джерел.....	51

## ВСТУП

За останні десятиліття кількість ТЗ у містах суттєво збільшилася, при цьому, їх транспортні системи не встигають розширюватися з такою ж швидкістю. Виникає не лише проблема із пробками, але й проблема із паркінгом. І хоча останнім часом цю проблему намагаються активно вирішувати, шляхом будівництва спеціалізованих платних парковок вона досі залишається основною проблемою. Особливо дана про проблема стає актуальною при будівництві нових, великих торгово-розважальних чи бізнес-центрів, у вже густо забудованих районах міста.

Потреба в паркувальних місцях визначається залежно від характеристик об'єкта, поряд з яким буде парковка (кількості одночасних відвідувань, кількості працівників, торгової площі та і т.д.). Як наслідок, неправильно організована парковка, в «час пік», може призвести до виникнення заторів та втрати відвідувачів, які хотіли відвідати об'єкт, але не змогли знайти місце для паркування.

Для вирішення цієї проблеми необхідно проектувати паркування, виходячи з характеристик об'єкта та інтенсивності руху на прилеглих дорогах. Розрахунок характеристик функціонування дозволив би заздалегідь визначити найбільш підходящий розмір території паркування, кількість заїздів і виїздів, їхнє найкраще розташування.

Мета роботи – розробити стохастичну модель ТП на парковці, реалізувати та виконати імітаційне моделювання на основі розробленої моделі.

Для досягнення мети виділено ряд завдань:

- побудова та опис стохастичної моделі КВ;
- розробка імітаційної моделі КВ;
- проведення імітаційного моделювання використовуючи отриману модель.



# 1 АНАЛІТИЧНА ЧАСТИНА

## 1.1 Опис предметної області

На даний момент, найбільш популярними підходами при проектуванні паркувань є методи статистичного аналізу. У роботі [1], розглядається процес проектування паркування під час побудови нового торгового центру. В ній використовується статистичний підхід, заснований на очікуваному попиті та розташування торгового центру. Також докладно розписуються способи організації самої парковки, що дозволяють більш детально описати модель, що працює.

Інший підхід представлений у роботі [2]. У ній вже використовуються імовірнісні характеристики, однак для вирішення проблеми та розрахунку оцінок ефективності паркування, пропонується скористатися низкою формул і рівнянь. Автори роботи представляють рішення у вигляді СМО, використовуючи інтенсивність попиту на паркування, як потік вхідних заявок. Проте, імітаційного моделювання в рамках цієї роботи зроблено не було.

Найточніших результатів можна досягти саме з допомогою імітаційного моделювання. У цьому можуть допомогти математичного моделювання та теорії ТП. Коректно побудована модель дозволить багаторазово моделювати стан паркування при різних значеннях параметрів. Тим самим дозволяючи отримати оптимальні параметри для конкретного паркування.

Перш ніж приступити до опису та реалізації моделі КВ, необхідно зрозуміти, який реально існуючий об'єкт моделюється, які його характеристики та які процеси з ним відбуваються. Крім усього іншого, необхідно зрозуміти, які вихідні дані надаються та як їх можна використати.

Саму модель можна розділити на два основні компоненти:

- КВ;
- ТП.

КВ будемо вважати спеціально відведене місце для стоянки ТЗ. Надалі замість терміна КВ буде використовуватися термін паркування. Парковка, по

суті, це ідеальний приклад КВ, дозволяє легко та доступно описати весь процес моделювання. Хоча, в теорії, види КВ не обмежуються тільки парковками або стоянками. До них також можна віднести дворові території, узбіччя або кінець транспортної мережі.

У свою чергу, ТП представляється як сукупність ТЗ, що рухаються якоюсь транспортною мережею. Моделювання ТП і транспортних мереж досліджується в рамках теорії ТП - науки, котра аналізує режими руху ТЗ у різних дорожніх умовах, з урахуванням їх динамічних можливостей, складу ТП та психофізіологічних особливостей водіїв [6, 7].

Транспортну мережу, де також є КВ, можна подати у вигляді транспортного графа (рис. 1.1), де червоним відзначені вершини, що позначають КВ, жовтим – паркування, а у вигляді дуг, виділених зеленим, позначаються дороги, якими здійснюють рух ТЗ.

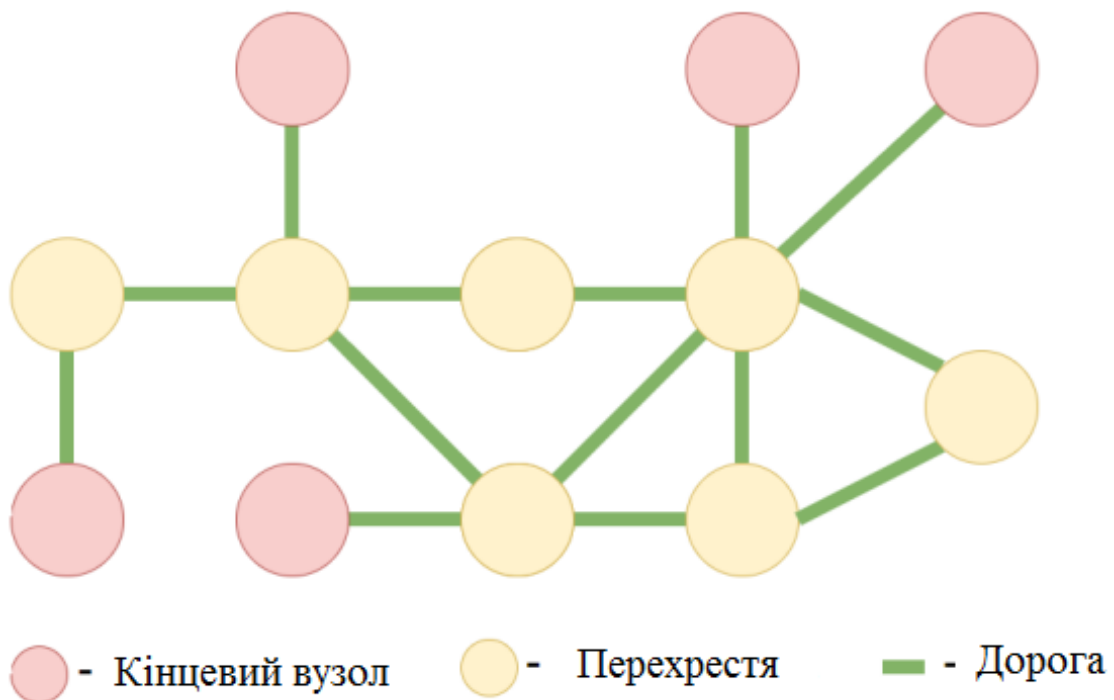


Рисунок 1.1 – Транспортний граф

Однак слід пояснити один важливий момент. При моделюванні паркування не розглядається весь ТП, а береться лише його частина. Іншими словами, з ТП, відповідно до ймовірнісних характеристик, вибирається якась

частина ТЗ, які мають наміри скористатися паркуванням. Потік таких ТЗ, визначається як ТП, що рухається в КВ.

При моделюванні ТП, що рухається в КВ, представляється як потік подій. Якщо бути точніше, то він репрезентується, як потік однорідних подій, що настають одна за одною у випадкові моменти часу. Надалі, такий потік подій і буде використано, як вхідний потік, для моделі, що розробляється.

## 1.2 КВ

Паркування призначене для стоянки ТЗ, які можуть в'їхати на неї з якоїсь ділянки транспортної мережі. Типова схема паркування представлена на рис. 1.2.

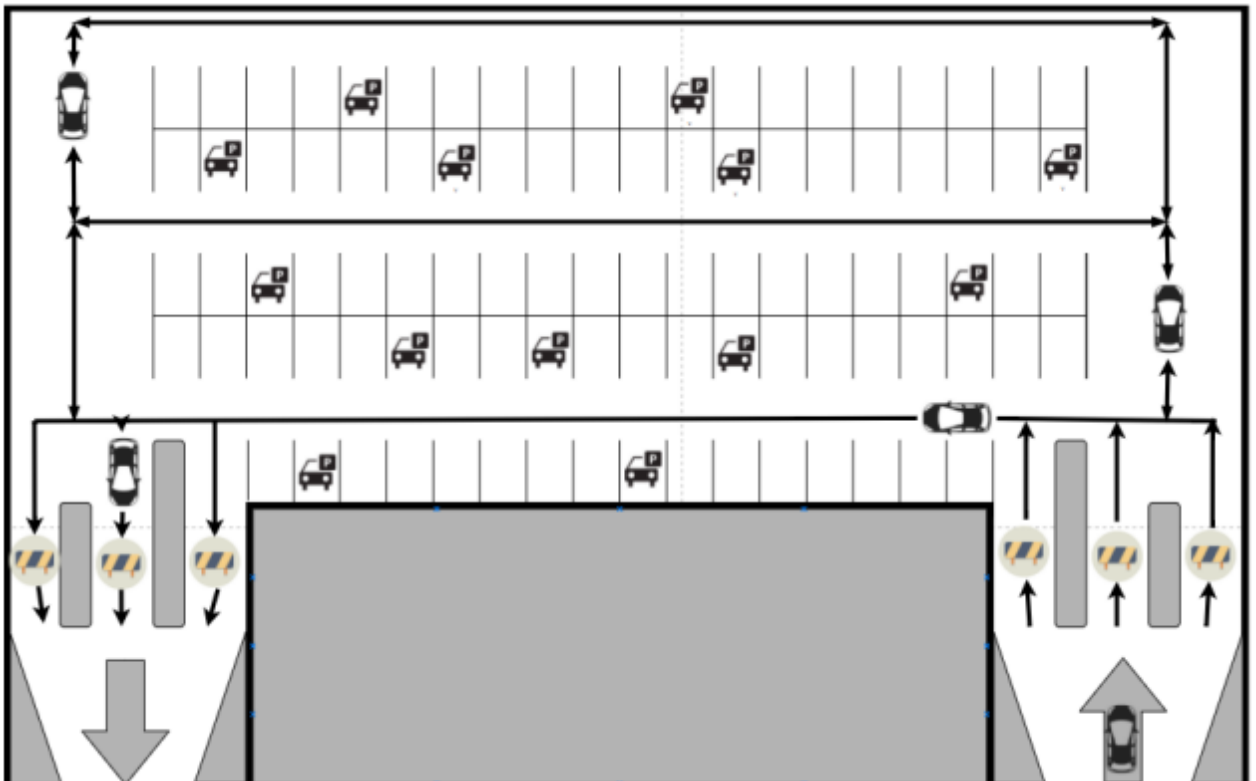


Рисунок 1.2 – Схема паркування

Весь процес проходження ТЗ через паркування можна розділити на 3 основні етапи:

- в'їзд;

- стоянка, що включає 2 основні дії: пошук місця для стоянки та стоянка;
- виїзд.

В'їзд на паркування починається із прилеглої дороги. При великій інтенсивності ТП, черга на в'їзд може починатися безпосередньо із даної ділянки транспортної мережі. Розмір цієї черги ніяк не обмежується, і вона може починатися з будь-якої ділянки міської транспортної мережі. На даному етапі, водії, що керують ТЗ ще можуть відмовитися від в'їзду на парковку і покинути дану чергу.

Безпосередньо поруч із парковкою, якщо вона має декілька заїздів, дорога може поділяється на кілька окремих, розділених перешкодами, найчастіше це бордюри, смуги руху. Іншими словами, якщо паркування має кілька в'їздів, то для кожного в'їзду виділяється своя смуга руху. Таким чином, ТЗ з одної великої черги розподіляються на декілька менших черг. На даному етапі залишити чергу не вдасться.

Самі ж в'їзди можуть бути обладнані шлагбаумами або іншими штучними перешкодами, призначеними для регулювання транспорту, що в'їжджає на паркування. Через це сам процес в'їзду може тривати деякий час.

Після в'їзду ТЗ потрапляє на саму територію паркування та переходить до наступного етапу - стоянки. Місця для стоянки мають більшу частину території паркування, проте, тут також є смуги для руху ТЗ. Кількість місць для стоянки ТЗ на парковці, обмежена.

На даному етапі, як було зазначено вище, відбувається 2 дії. Перша – водій шукає місце для стоянки. Здійснюється це шляхом руху по виділених смугах. Як наслідок, тут можливі два результати – водій знайшов місце для стоянки і водій не знайшов місце для стоянки. При настанні успішного результату виконується друга дія, описана для даного етапу, і ТЗ ставитися на стоянку. Після стоянки, яка триває деякий час, ТЗ переходить до наступного етапу - виїзду.

При невдачі, ТЗ одразу переходить до наступного етапу – виїзду та

залишає паркування. Етап виїзду аналогічний в'їзду, змінився тільки напрямок руху ТЗ. Сам виїзд починається безпосередньо з території паркування. Якщо бажаючих залишити територію багато, то вони утворюють загальну чергу, займаючи смуги, призначені для руху ТЗ по території самої парковки. Ближче до виїзду з парковки, якщо вона має кілька виїздів, дорога може поділяється на кілька окремих, розділених якоюсь перешкодою, смуг руху. Іншими словами, якщо паркування має кілька виїздів, то для кожного виїзду виділяється своя смуга руху.

Як і в'їзд, виїзд може бути обладнаний шлагбаумами або іншими штучними перешкодами, призначеними регулювання руху ТЗ через парковку. Після виїзду ТЗ виїжджає на ділянку прилеглої дороги і продовжує рух міською транспортною мережею.

### **1.3 ТП, що рухається в КВ**

ТП, це сукупність ТЗ, котрі рухаються проїжджою частиною дороги, і мають намір проїхати через КВ. Основними характеристиками ТП є:

- інтенсивність;
- щільність;
- швидкість.

ТП є невід'ємною частиною моделі, що розробляється у межах даної роботи. Від його характеристик і залежить те, яка поведінка буде біля парковки. Основною характеристикою ТП, яка використовується в рамках цієї роботи, є інтенсивність. Інтенсивність руху визначається числом ТЗ, котрі рухаються у певному напрямку по дорозі і проходять через пункт спостереження за фіксований період часу [2, 8]. Знаючи інтенсивність руху можна визначити, скільки ТЗ, в середньому, проїжджатиме через парковку за годину, день або місяць.

При цьому ТП не є стаціонарним і змінюється з плином часу. Як наслідок, кількість ТЗ, котрі заїжджають на парковку, також змінюється з часом. Через це в один момент часу парковка може бути порожньою, а через

годиною вже може утворитися затор.

#### **1.4 Опис моделі, що розробляється**

Для проведення імітаційного моделювання КВ необхідно розробити дві моделі:

- модель ТП, що в'їжджає на КВ;
- модель КВ.

Модель ТП пропонується у вигляді потоку подій. Це має бути потік однорідний подій, що наступають один за одним у довільні моменти часу.

Розроблювану стохастичну модель КВ пропонується у вигляді багатофазної СМО, у якій кожній фазі відповідає своя роль - в'їзд, стоянка та виїзд.

##### **1.4.1 Модель вхідного потоку**

Як згадувалося раніше, вхідний потік буде представлений як потік однорідних подій, що настає один за одним у довільні моменти часу. Використовуючи подібний потік буде враховано головну властивість ТП - кількість ТЗ, що в'їжджають для паркування змінюється з часом.

Модель вхідного потоку також повинна залежати від середньої інтенсивності руху ТЗ, що рухаються прилеглою дорогою.

Для того, щоб виконати обидві ці умови і змоделювати вхідний потік, було вирішено використовувати нестационарний пуассонівський потік. Цей потік залежить від певної максимальної інтенсивності і дозволяє створювати події у випадкові моменти часу

Вихідними даними, для моделі вхідного потоку, є інформація про інтенсивність руху ТЗ по ділянці транспортної мережі, що прилягає до паркування. Щодо інформації про інтенсивність протягом усього дня, вираховується інтенсивність руху ТЗ по годинах. Для цього використовуються коефіцієнти, що являють собою інтенсивність руху, у відсотках від добової.

Приклад отримання таких коефіцієнтів описаний у роботі [2].

У результаті, після всіх розрахунків, виходить гістограма, показана на рис. 1.3. Вона відображає середню кількість ТЗ, що проїжджають через парковку за конкретну годину.

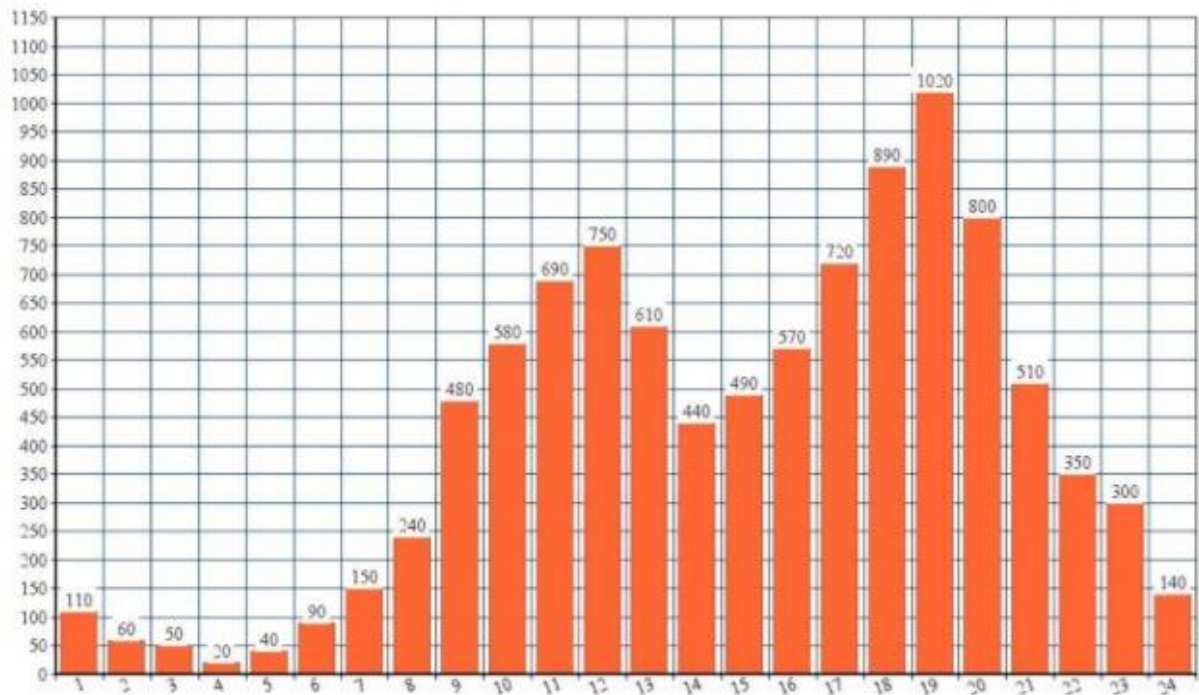


Рисунок 1.3 – Добова інтенсивність руху через парковку, з поділом по годинах

Вхідний потік для моделі КВ представлений як потік подій. Потік подій можна описати як послідовність певних подій, котрі настають у випадкові моменти часу.

Найпростіший (стаціонарний пуассонівський) потік - потік подій, який має властивості стаціонарності, відсутності наслідку та ординарності.

Нестаціонарний пуассонівський потік - потік подій, що володіє властивостями ординарності та відсутності наслідку, але не є стаціонарним.

Детальніше про самі характеристики:

– властивість стаціонарності: ймовірність появи до подій на будь-якому проміжку часу залежить тільки від числа до  $i$  від тривалості  $t$  проміжку і не залежить від початку його відліку;

– властивість ординарності: ймовірністю наступу за елементарний проміжок часу більше однієї події можна знехтувати порівняно з ймовірністю наступу за цей проміжок не більше однієї події (тобто ймовірність одночасної появи двох і більше подій дорівнює нулю);

– властивість відсутності наслідку: ймовірність появи до подій на будь-якому проміжку часу не залежить від того, з'являлися або не з'являлися події в моменти часу, що передують початку розглянутого проміжку.

Для моделювання нестационарного потоку буде використаний метод проріджування, детальна реалізація якого буде розглянута в розділі 3 даної роботи [3, 9]. Основна ідея алгоритму полягає в тому, що спочатку ділиться стаціонарний пуасонівський потік із заданою максимальною інтенсивністю, а потім, з деякою ймовірністю, частина подій буде відкинута. Отже, будуть забезпечені всі основні властивості ТП.

#### **1.4.2 Модель КВ**

Представлена у вигляді трифазної СМО, де кожна фаза відображає певний етап проїзду через паркування. Заявки, що залишили фазу в'їзду, одразу потрапляють на вхід фази стоянки. Заявки, що залишили фазу стоянки, одразу потрапляють на вхід фази виїзду. На вхід даної моделі подається потік заявок, отриманий за допомогою моделювання ТП.

Докладно буде описана в розділі 2.



## 2 ТЕОРЕТИКО-ПРАКТИЧНА ЧАСТИНА

### 2.1 Введення в теорію СМО

У загальному випадку СМО може бути представлена моделлю, котра містить такі основні компоненти [10] (рис. 2.1).

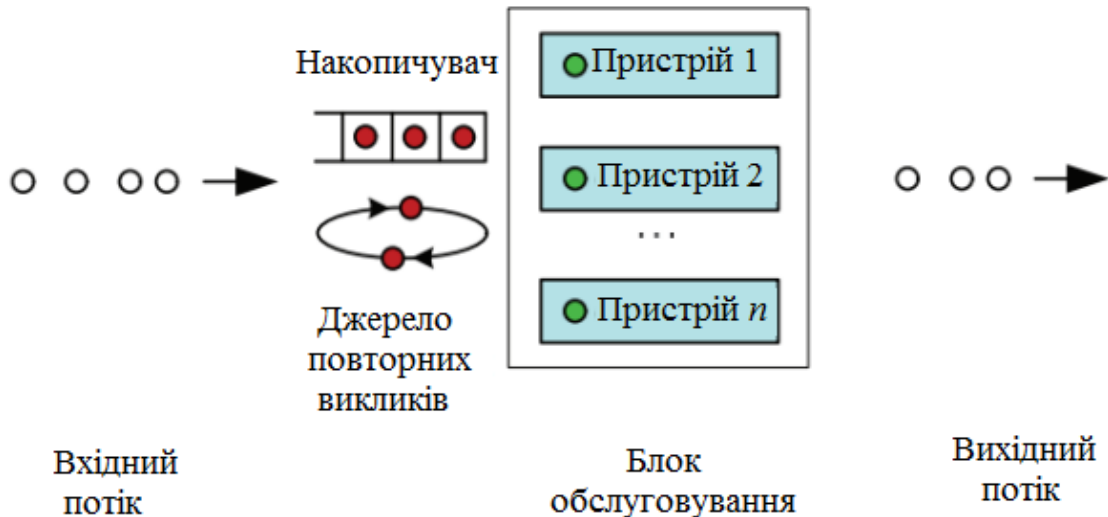


Рисунок 2.1 – Загальна модель СМО

На рисунку зображені такі основні елементи СМО:

- вхідний потік об'єктів, які називаються різними авторами по-різному: заявками, запитами, клієнтами; у межах цієї роботи - потік ТЗ;
- блок обслуговування (один або декілька пристроїв обслуговування); у межах цієї роботи - в'їзди, виїзди, місця для стоянки;
- накопичувач, у якому розміщується черга;
- вихідний потік об'єктів.

СМО може представлятися у вигляді багатофазної моделі, коли кожна вимога в ній послідовно обслуговується у всіх фазах (пристроях обслуговування). Загальна схема трифазної СМО наведено на рис. 2.2.

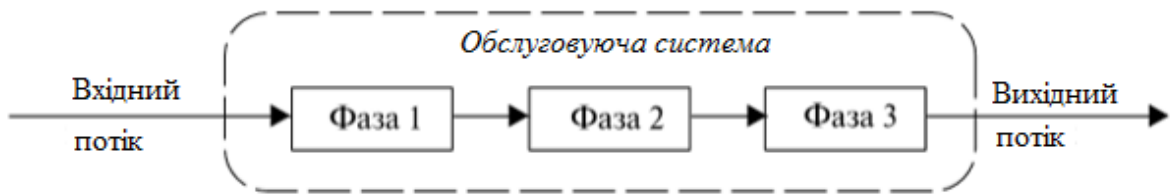


Рисунок 2.2 – Приклад трифазної СМО

Повна схема моделі КВ, що розробляється, представлена на рис. 2.3. На схемі показані всі елементи моделі.

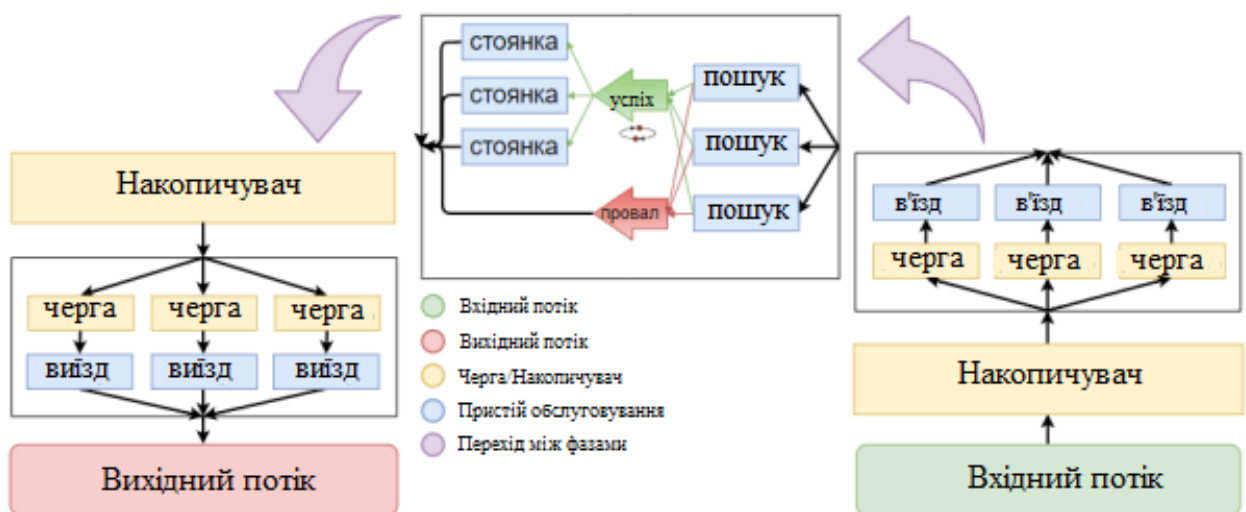


Рисунок 2.3 – Повна схема моделі КВ

### 2.1.1 СМО для моделювання в'їзду на КВ

Першою фазою, на яку надходить потік заявок, є в'їзд на паркування. Ця фаза відображена як СМО  $G/G/n/\infty$  (див. рис. 2.4). Де перша  $G$  позначає довільний вхідний потік, друга  $G$  - довільний час обслуговування.

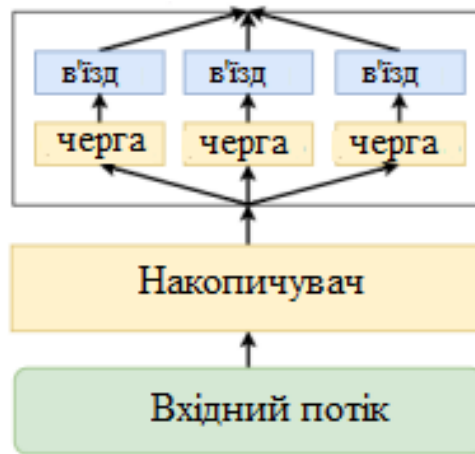


Рисунок 2.4 – СМО для моделювання в'їзду

Для даної СМО використовується комбінована черга, котра містить дві частин. Перша з них є повнодоступною чергою, з нетерплячими клієнтами, куди потрапляють усі заявки без винятку. Розмір цієї черги необмежений. Нетерплячість клієнтів визначається або параметром, що задає інтенсивність залишення черги, або функцією розподілу ймовірностей часу перебування в черзі, до моменту її залишення. Друга частина комбінованої черги на в'їзд - це  $n$  неповнодоступних черг, де  $n$  - кількість обслуговуючих пристроїв (в'їздів), кожна черга, закріплена за конкретним обслуговуючим пристроєм. Клієнти в черзі – терплячі, і не покидають її. Розмір черг задається окремим параметром.

В'їзд виконується завдяки обслуговуванню на пристроях. Власне час обслуговування описується деякою заданою функцією розподілу. Після обслуговування, всі заявки надходять на вхід фази 2.

### 2.1.2 СМО для моделювання стоянки на КВ

Основним етапом описуваної моделі є фаза стоянки (фаза 2). Ця фаза представлена як двофазна СМО  $\cdot/G/\infty \rightarrow \cdot/G/n/0$  (рис. 2.5).

Тут передбачено дві фази обслуговування – пошук вільного місця та безпосередньо, стоянка. Спочатку заявка, що надійшла в цю систему, потрапляє для обслуговування на фазу пошуку, тим самим моделюючи поведінку, коли Авт, що в'їхав на паркування, вибирає (шукає) місце для стоянки. Після того, як місце вибрано, він стає на стоянку, тобто заявка переходить на обслуговування

до другої фази  $\cdot/G/n/0$ . Тут черга відсутня, а кількість обслуговуючих пристроїв  $n$  – це кількість місць для стоянки Авт.

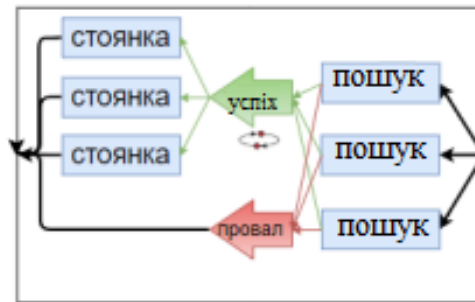


Рисунок 2.5 – СМО для моделювання стоянки

Якщо ж після часу пошуку (перебування на першій фазі), клієнт не зміг знайти вільний пристрій для стоянки (на другій фазі), він прямує до виїзду, тобто заявка відправляється на вхід фази 3 СМО паркування. Також заявка потрапляє на вхід фази 3 СМО паркування та після завершення стоянки.

### 2.1.3 СМО для моделювання виїзду з КВ

Заключним етапом є фаза виїзду (фаза 3). Всі заявки, що пройшли через фазу 1 і фазу 2, успішно чи ні - не має значення, попадають цей етап. Дана фаза представлена, як СМО  $\cdot/G/n/\infty$  (рис. 2.6) і є повним аналогом фази 1 за одним винятком - на даній фазі неможливі втрати і клієнти не залишають чергу.

Всі клієнти, що закінчили обслуговування на фазі 2, спочатку потрапляють до комбінованої черги фази 3. Ця черга, як і на фазі 1, складається з двох частин. Перша частина - повнодоступна черга, з терплячими клієнтами, на якій не можливі втрати. Розмір цієї черги не обмежений. Друга частина комбінованої черги - це  $n$  неповнодоступних черг, де кожна черга закріплена за конкретним виїздом (пристроєм обслуговування). Виїзд здійснюється за допомогою обслуговування на пристроях. Час обслуговування визначається певною функцією розподілу.

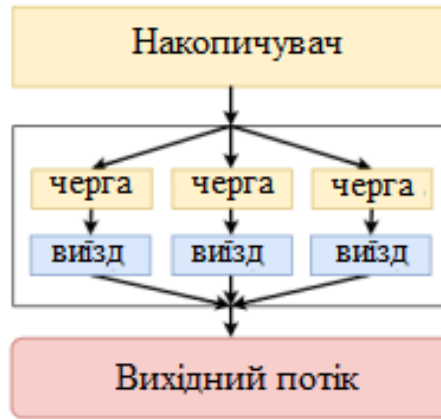


Рисунок 2.6 – СМО для моделювання виїзду

Після того, як заявка закінчила обслуговування на фазі виїзду, вона кидає всю систему.

#### 2.1.4 Час обслуговування

На кожній фазі проїзду через паркування є пристрої обслуговування різних типів. Для того, щоб задати час обслуговування на пристроях, не використовується конкретний розподіл. Для моделювання можна задати будь-який зручний для користувача метод.

### 2.2 Практична реалізація імітаційної моделі

У рамках цієї роботи було реалізовано додаток, що дозволяє моделювати проходження ТЗ через паркування. Для генерації вхідного потоку використовується нестационарний пуасонівський потік, реалізований за допомогою методу проріджування. Для реалізації моделі КВ використано Python фреймворк Salabim. Результатами моделювання є статистичні та імовірнісні характеристики, представлені в текстовому та графічному вигляді.

#### 2.2.1 Фреймворк Salabim

Salabim - це Python фреймворк процесоорієнтованої, дискретно-подійної

системи моделювання [5].

Процеси Salabim визначаються за допомогою функцій-генераторів з Python і можуть, наприклад, використовуватися для моделювання активних компонентів, таких як клієнти, ТЗ або агенти. Salabim також надає різні типи загальних ресурсів, які використовуються для моделювання обмежених пропускних пунктів, таких як сервери, пункти контролю, тунелі [3, 11].

Моделювання може виконуватись у трьох режимах:

- "as fast as possible" (моментальне моделювання);
- "real time" (в режимі реального часу);
- перехід за подіями вручну.

Теоретично, в Salabim можна зробити безперервну симуляцію, але практично це неможливо. Однак, у випадку моделювання з кроком фіксованого розміру, де процеси не взаємодіють один з одним або із загальними ресурсами, може використовуватися простий цикл while.

Усі процеси (processes) "живуть" в оточенні (environment). Вони взаємодіють із оточенням і один з одним, у вигляді подій (events).

Процеси описуються простими функціями-генераторами з Python. Їх можна використовувати як функцію процесу або метод процесу, залежно від того, чи є він нормальною функцією або методом класу. Протягом свого «життя» вони створюють події та передають (yield) їх в оточення, очікуючи на їх спрацювання. Коли процес передає подію, він припиняє свій потік виконання. Salabim відновлює процес, коли настає подія. При цьому можливий випадок, коли кілька процесів можуть очікувати настання однієї і тієї ж події. У цьому випадку Salabim відновлює процеси у тому порядку, в якому вони передали цю подію.

Події бувають кількох типів. Найбільш використовуваним типом події є Timeout. Події такого типу призупиняють процес і відновлюють його після того, як минула певна кількість часу. Іншими словами, Timeout дозволяє процесу утримувати свій стан протягом певного часу, який задається при породженні цієї події. Timeout і будь-які інші події можуть бути створені шляхом виклику відповідного методу з оточення, в якому знаходиться цей процес, наприклад ,

`Environment.timeout()` '.

Для того, щоб остаточно показати будову фреймворку, розглянемо невеликий і простий приклад, який моделює рух ТЗ. Лістинг 2.1 відображає функцію, що описує процес під назвою «car». Для створення подій цього процесу потрібне посилання на оточення («env»), в рамках якого він буде виконуватися. Поведінка ТЗ описується у нескінченному циклі. Ця функція є генератором. Хоча вона ніколи не завершиться, але при досягненні ключового слова «yield» вона передасть потік управління в симуляцію. Після того, як передана подія настане, симуляція буде продовжена з цього ж місця.

Лістинг 2.1 – Приклад функції, що описує процес руху Авт

```
>>> def car(env):
...     while True:
...         print('Start parking at %d' % env.now)
...         parking_duration = 5
...         yield env.timeout(parking_duration)
...
...         print('Start driving at %d' % env.now)
...         trip_duration = 2
...         yield env.timeout(trip_duration)
...
... 
```

У цьому прикладі ТЗ переходить між двома станами - стоянка і водіння. Процес оголошує про свій новий стан, друкуючи повідомлення та поточний час моделювання (отриманий при зверненні до методу `Environment.now()`). Потім він викликає фабричну функцію `Environment.timeoutQ` для створення події з типом `Timeout`. Ця подія визначає момент часу, коли Авт здійснює парковку (або водіння, відповідно). Передавши подію, він сигналізує симуляції, що хоче дочекатися настання цієї події.

Після того, як поведінку нашого Авт було описано, можна створити його екземпляр і подивитися, як він поведеться (лістинг 2.2).

Перше, що потрібно зробити, це створити екземпляр оточення. Цей екземпляр передається в описану вище функцію при виклику у вигляді параметра. Виклик цієї функції повинен бути доданий в оточення за допомогою методу `'Environment.process()'`.

## Лістинг 2.2 – Приклад запуску симуляції

```
>>> import salabim
>>> env = salabim.Environment()
>>> env.process(car(env))
<Process(car) object at 0x836dff0>
>>> env.run(until=15)
Start parking at 0
Start driving at 5
Start parking at 7
Start driving at 12
Start parking at 14
```

Варто зауважити, що в даний час жоден з кодів функції процесу не виконується. Його виконання просто заплановане зараз моделювання.

Об'єкт типу `Process`, отриманий при виклику однойменної функції `'process'`, може використовуватися для взаємодії з іншими процесами.

Після всього виконаного необхідно запуснути симуляцію, викликавши метод `'run'`, передавши йому час закінчення моделювання.

### 2.2.2 Основні компоненти програми

`Salabim` надає інтерфейс для моделювання за допомогою спеціального набору класів, де кожен клас має своє призначення. За допомогою взаємодії екземплярів цих класів і здійснюється процес моделювання.

Список класів, що надаються фреймворком і використовуються в додатку:

- `Environment` - клас, за допомогою якого здійснюється взаємодія всіх створених процесів і за допомогою якого здійснюється сам процес моделювання.
- `Process` - клас, що дозволяє описати моделюваний об'єкт та визначити його поведінку.
- `Queue` - клас, що використовується для опису черг до ресурсів або інших об'єктів моделі.
- `Resource` - клас, який використовується для опису ресурсів, до яких звертається об'єкт, що моделюється.
- `Monitor` - Клас, за допомогою якого здійснюється збір статистичної інформації про моделювання. Цей клас використовується всіма об'єктами.



Для реалізації, описаної в розділі 2 моделі, було вирішено використати наступний набір компонентів:

- process: ТЗ, в'їзд та виїзд;
- resource: пошук та стоянка;
- queue: загальні черги, черги на в'їзд та виїзд.

Обновивши рисунок 2.3, підставивши використовувані компоненти місця об'єктів, які вони моделюють, отримаємо схему, показану на рис. 2.7.

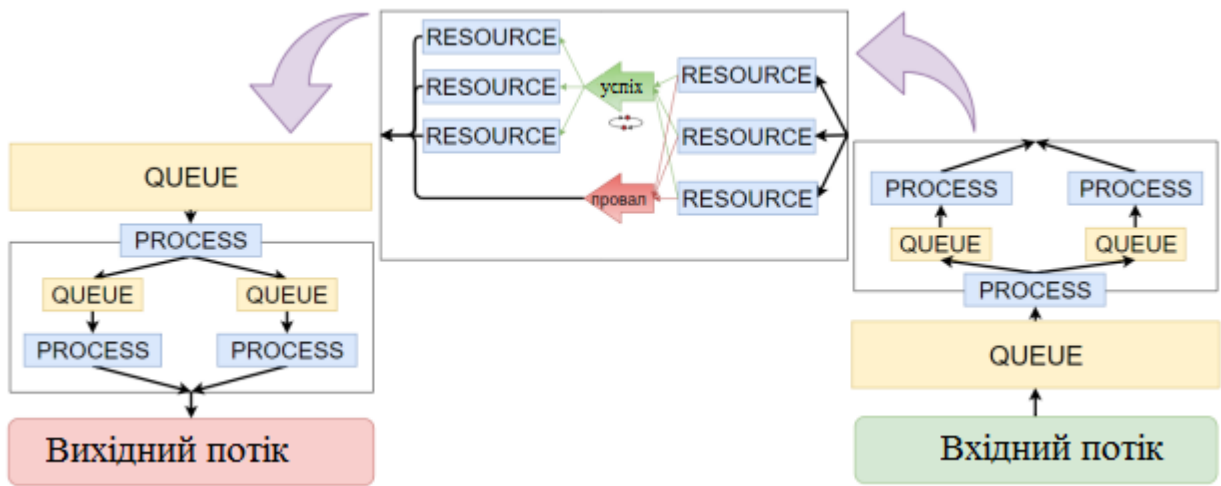


Рисунок 2.7 – Схема компонент, які використовуються та зв'язку між ними

Список методів, що використовуються при розробці програми та їх опис наведено нижче:

- hold - метод, який зупиняє виконання процесу на який, певний час (на вхід методу подається час у секундах);
- passivate - метод, який зупиняє виконання процесу на деякий, невизначений час (є аналогом hold, тільки в даному випадку час вважається рівним нескінченності);
- activate - метод, що відновлює виконання процесу;
- pow - метод, який повертає поточний час моделювання;
- cancel - метод, що завершує виконання процесу негайно ;
- enter - метод, що дозволяє додати екземпляр класу, який моделює певний процес, у чергу (на вхід подається ім'я дуже рідко; метод викликається з

процесу);

– leave - метод, що дозволяє видалити екземпляр класу, який моделює якийсь процес, з черги (на вхід подається ім'я черги; метод викликається з процесу);

– request - метод, що дозволяє звернутися до ресурсу, та забрати його собі (на вхід подається ім'я ресурсу);

– release - метод, що дозволяє звільнити ресурс, який було забрано раніше (на вхід подається ім'я ресурсу).

Вхідний потік (метод розрідження). Вхідний потік представлений у вигляді нестационарного пуассонівського потоку, що генерується за допомогою алгоритму розрідження. Блок схема даного алгоритму представлена рис. 2.8.

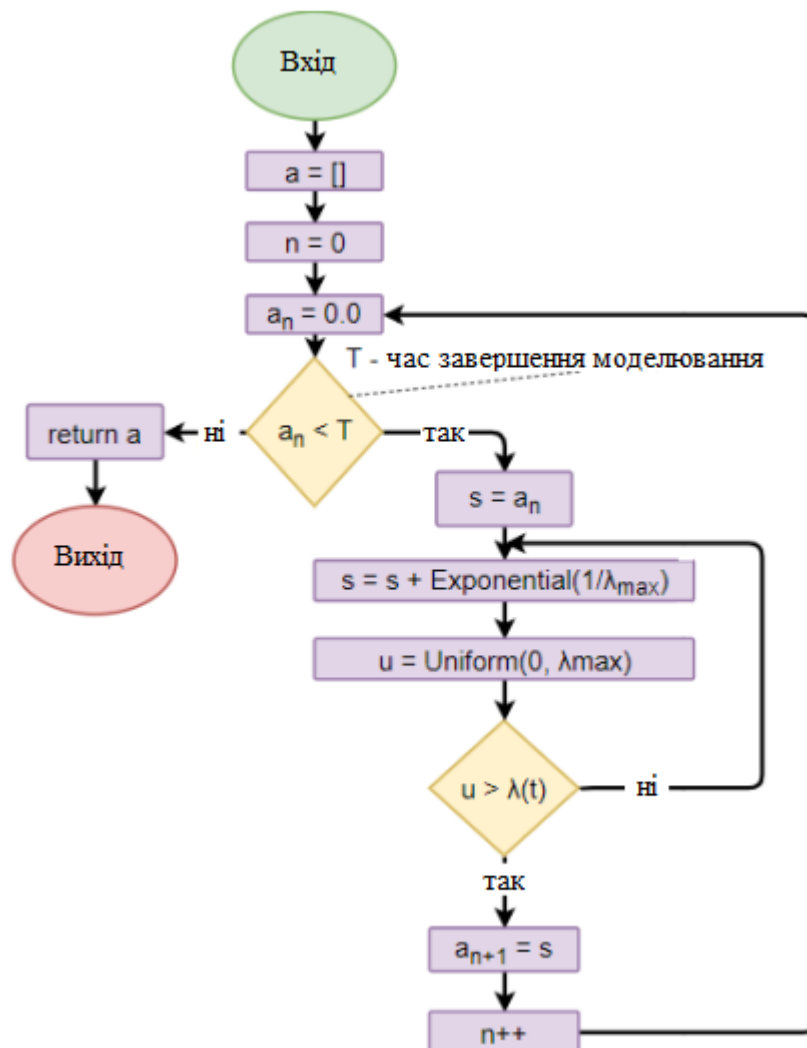


Рисунок 2.8 – Блок-схема алгоритму розрідження

Цей алгоритм складається з кількох послідовних кроків. На вхід алгоритму подається час закінчення моделювання. Поки цей час не досягнуто, алгоритм продовжує свою роботу та генерує нові моменти часу.

Новий момент часу, тобто час, коли заявка надійшла на вхід моделі, генерується таким чином:

- береться поточний час і надається змінної Б;
- поки що випадкова величина  $i$ , більше, ніж значення інтенсивності від часу Б, виконуємо:
  - до часу Б додаємо значення експоненціального розподілу, отриманого на основі максимальної інтенсивності ;
  - обчислюємо випадкову величину  $i$ , як випадкова величина, отримана за допомогою рівномірного розподілу , від 0 до максимальної інтенсивності;
  - додаємо в масив результатів нові моменти часу Б.

В результаті роботи алгоритму на виході маємо масив з моментів часу, згідно з яким у кожен момент часу з'являється нова заявка, або, у разі моделі, що розробляється - ТЗ.

Реалізація алгоритму і результат його роботи показаний в лістингу 2.3.

### Лістинг 2.3 – Реалізація алгоритму розрідження

```
>>> def thinning_method() -> List[float]:
...     a = [0.0]
...     n = 0
...     while a[n] < _day:
...         s = a[n] + exponential(_hour / _max_rate)
...         u = uniform(0, _max_rate)
...         while u > _lambda(s):
...             s += exponential(_hour / _max_rate)
...             u = uniform(0, _max_rate)
...         a.append(s)
...         n += 1
...     return a
...
>>> thinning_method()
[0.0, 24.814140030607877, 28.80060783734734, ..., 86425.2015097532]
```

Тепер після того, як у нас з'явився масив з часів, ми можемо почати створювати самі заявки. Для цього використовуватиметься компонент

фреймворку Salabim - Process. Таким чином, буде описаний процес, який моделює прибуття ТЗ на паркування (лістинг 2.4). У самих ТЗ буде вже свій процес, що описує проходження одного Авт через все паркування.

#### Лістинг 2.4 – Генератор ТЗ

```
class CarGenerator(Process):
    def process(self):
        times = thinning_method()
        prev = 0
        for time in times:
            Car()
            yield self.hold(time - prev)
            prev = time
```

Опис ТЗ. Основним елементом моделі, що розробляється, є процес, що описує поведінку ТЗ при проїзді через паркування. Як було показано раніше, саме ТЗ створюється у певний момент часу. ТЗ описується за допомогою класу Process. Після того, як було створено об'єкт цього класу, оточенням викликається метод 'process'. Вся логіка поведінки ТЗ при проїзді через паркування, якраз і описана в даному методі.

Таким чином, в даному методі описано взаємодію ТЗ з усіма елементами моделі. За допомогою цієї взаємодії відбувається процес моделювання.

Далі буде докладно розглянуто, як моделюється рух ТЗ через усі етапи паркування. Лістингу 2.5 містить метод 'process', що описує поведінку ТЗ. На даному лістингу проходження кожної фази виділено відповідним кольором. Зеленим виділено в'їзд на парковку, синім – стоянка, а червоним відзначено виїзд. Усі зайві методи, пов'язані зі збором статистики, були опущені. Щодо даного методу і буде будуватися подальше пояснення.

## Лістинг 2.5 – Клас, що описує поведінку ТЗ на парковці

```
class Car(Process):
    def __init__(self, *args, **kwargs):...

    def process(self):
        if len(input_shared_queue) > max_length_of_input_queue:
            yield self.cancel()

        self.enter(input_shared_queue)
        yield self.hold(exponential(mean_wait_time_for_input_shared_queue))
        if self in input_shared_queue:
            if input_shared_queue.index(self) > max_car_position:
                self.leave(input_shared_queue)
                yield self.cancel()
            else:
                yield self.passivate()

        yield self.hold(exponential(mean_search_time))
        if parking.available_quantity() != 0:
            yield self.request(parking)
            yield self.hold(exponential(mean_parking_time))
            self.release()
        else:
            pass

        self.enter(output_shared_queue)
        yield self.passivate()
```

В'їзд на КВ. У певний момент часу, на прилеглий до паркування дорозі, з'явився новий ТЗ, який бажає скористатися цією самою парковкою. В рамках даної програми це означає, що був створений новий екземпляр класу Car і був викликаний метод 'process'.

Перш за все, розглянемо, які дії може здійснити водій при в'їзді на парковку. Приклад того, як може діяти водій під час в'їзду на парковку, представлений на рис. 2.9.

Насамперед, водій ТЗ, дивиться на кількість ТЗ, що стоять у загальній черзі. Якщо в черзі Авт більше, ніж він очікує побачити, він відмовляється від в'їзду і залишає прилеглу дорогу. Якщо ТЗ у черзі не багато або їх взагалі немає, то водій приймає рішення скористатися паркуванням та встає у чергу. Якщо, через якийсь час, водій бачить, що він досі не в'їхав, і він ще далеко від в'їзду, то він залишає чергу. Інакше, якщо в'їзд вже поряд, він залишається у черзі і чекає на момент виїзду

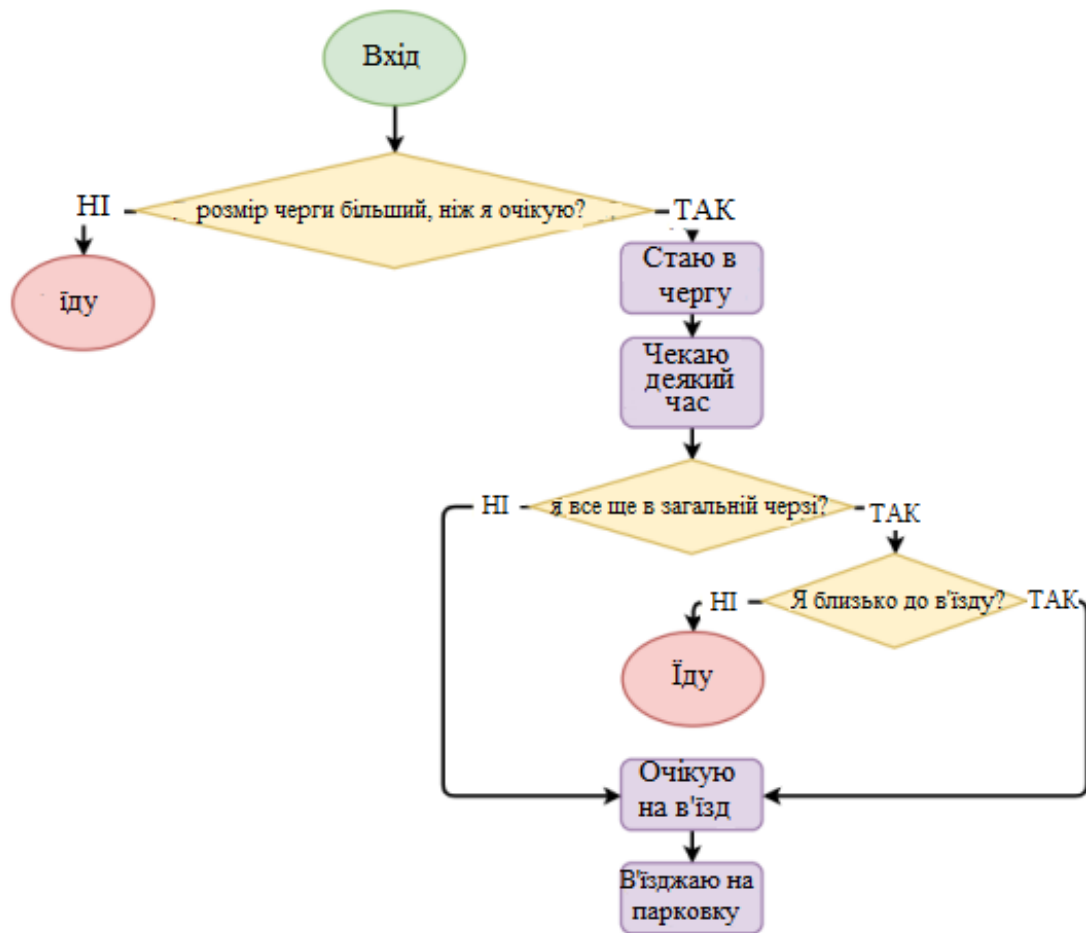


Рисунок 2.9 – Поведінка водія на фазі 1

Тепер розглянемо, як ці дії виконуються в коді додатка. Після виклику методу 'process' для нового ТЗ, виконується умова, згідно з якою перевіряється, чи не є вхідна, загальна черга занадто великою. Якщо розмір черги великий, викликається спеціальний метод 'cancel', який зупиняє виконання процесу і видаляє об'єкт класу з оточення. Після виклику даного методу управління до процесу вже не повернеться. Якщо розмір черги не великий або черга порожня, то об'єкт класу додається до кінця загальної черги. Після того, як об'єкт класу був доданий до кінця загальної черги, процес стає неактивним на деякий час, передавши потік управління в оточення і попросивши його повернути через N секунд. Або поки хтось не активує цей процес повторно. Робиться це за допомогою спеціального методу 'hold', який робить процес пасивним на деякий час, передане при виклику цього методу. Щойно управління повернули, перевіряється умова, що об'єкт класу залишив загальну чергу. Якщо умова не

виконується, це означає, що управління повернуто по таймауту. Якщо об'єкт класу все ще у спільній черзі, то перевіряємо, наскільки він далекий від початку цієї загальної черги.

Залежно від того, які умови виконані, а які ні, можливі 3 варіанти дії:

- закінчуємо виконання процесу та видаляємо об'єкт класу з оточення (метод 'cancel');
- процес повторно робиться неактивним, але вже без таймауту; очікуємо, поки не настане момент в'їзду, для ТЗ (метод 'passivate');
- переходимо до виконання наступної фази (виконується наступний блок коду, що відповідає за моделювання фази 2).

Однак залишилося ще згадати один важливий момент. Як екземпляр класу переходить з однієї черги до іншої, і те, як додаток розуміє, що було виконано в'їзд.

Почнемо з процесу, що моделює сам процес в'їзду ТЗ на стоянку. В'їзд представлений у вигляді окремого процесу, котрий забирає один, перший об'єкт класу з черги, обслуговує його та повертає управління (активує) цей процес. Таких процесів, що моделюють в'їзд ТЗ на стоянку, може бути кілька, у кожного процесу своя черга і вони можуть розділитися за типом. Наприклад, одні використовують для моделювання в'їзду, інші для моделювання виїзду

Стоянка на КВ. Наступним етапом є стоянка ТЗ. Вона поділяється на два підетапи: пошук місця для стоянки і безпосередньо сама стоянка. У додатку дані дії слідує відразу ж за блоком коду, що відповідає за в'їзд ТЗ на парковку. Можлива поведінка на даній фазі показано на рис. 2.10.

Перший етап фази стоянки моделюється досить простим способом. Для того, щоб показати, що водій витратив деякий час на їзду територією стоянки, використовується метод 'hold'. Таким чином моделюється те, що водій, перш ніж поставити Авт на стоянку, витратив деякий час, перш ніж знайти це саме місце. Час очікування, який вдається при виклику методу, розраховується виходячи з поточної завантаженості стоянки. Чим більше зайнято місць, тим більше часу пошуку.

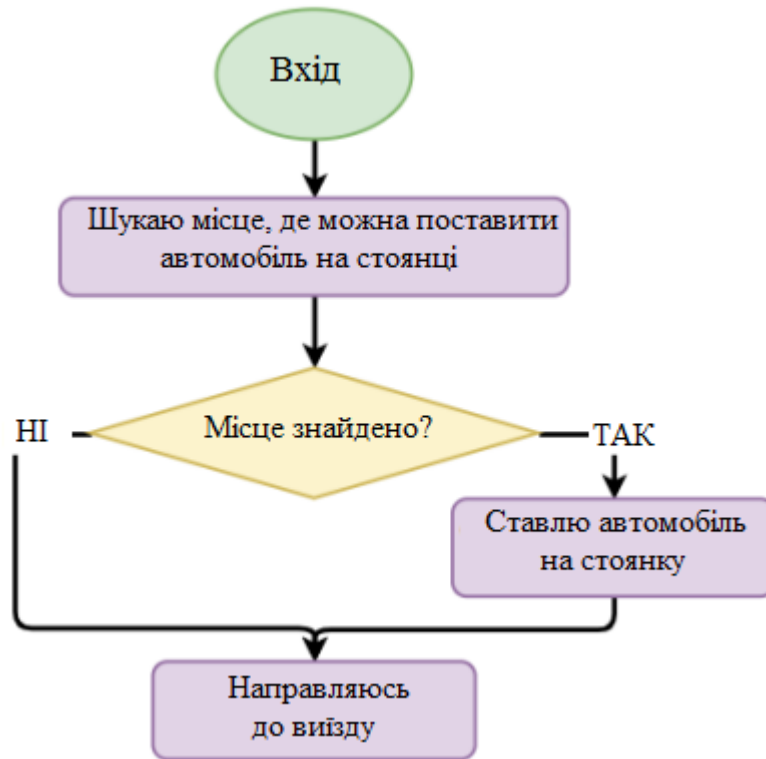


Рисунок 2.10 – Поведінка водія на фазі 2

Для моделювання другого етапу використовується об'єкт класу типу Resource (ресурс). Кількість даного ресурсу відповідає кількості місць на стоянці та визначається параметром capacity (лістинг 2.7). На ньому також показано момент створення спільних черг для в'їзду та виїзду. Посилання на створені об'єкти є у кожного об'єкта типу Car.

Лістинг 2.7 – Приклад створення об'єктів типу Resource та Queue

```

parking = salabim.Resource(name="parking", capacity=parking_size)
input_shared_queue = salabim.Queue(name="input_shared_queue")
output_shared_queue = salabim.Queue(name="output_shared_queue")

```

Для того, щоб дізнатися кількість вільних місць, використовується один із спеціальних методів - `available\_quantity` або `claimed\_quantity`. Вони відображають кількість доступних і кількість використовуваних ресурсів відповідно. Якщо вільні місця є, то для того, щоб забрати одне місце, слід використати метод 'request', при виклику якого у вигляді параметра необхідно передати ім'я ресурсу, який необхідно зарезервувати. Після того, як ресурс став



непотрібним, його необхідно повернути за допомогою методу 'release' , вказавши при виклику ім'я ресурсу, який ми повертаємо назад.

Таким чином, весь етап стоянки зводиться до наступного виклику методів:

- hold (моделюємо момент пошуку місця на стоянці; водій їздить територією стоянки);
- available\_quantity або claimed\_quantity (для того, щоб дізнатися, чи є вільні місця на стоянці);
- request (для того, щоб стати на місце для стоянки);
- hold (моделюємо стоянку ТЗ на парковці протягом якогось часу);
- release (для того, щоб звільнити місце для стоянки)

Виїзд КВ. Як мовилося раніше попередніх розділах, фаза 3 - виїзд, є повним аналогом фази 1 - в'їзд. Тому набір об'єктів, що використовуються для моделі цієї фази, такий же, як і для фази 1. Проте, так як на розмір загальної черги вже можна не звертати уваги, іншими словами, клієнти тут терплячі, то можна прибрати додаткові умови, перевірка котрих відбувається при моделюванні в'їзду ТЦ на територію стоянки.

Поведінка водія досить проста, він встає в чергу і чекає, момент, коли він зможе виїхати з території паркування. Іншими словами, коли він доїде до виїзду. Його можлива поведінка зображена на рис. 2.11.

Таким чином програмний код, що використовується для моделювання виїзду, вмістився у два рядки. Спочатку об'єкт класу додається у загальну чергу на виїзд, після чого він стає пасивним, передавши потік виконання в оточення на невизначений час і чекає, коли йому його повернуть. Обробка у чергах повністю ідентична фазі 1. Його спершу переміщає із загальної черги до черги виїзду. Робить це спеціальний компонент QueueManager. Після чого, як доходить його черга на виїзд, він обслуговується на компоненті Gate. Після того, як об'єкт класу був обслужений, екземпляр класу Gate його активує, повернувши управління потоком.

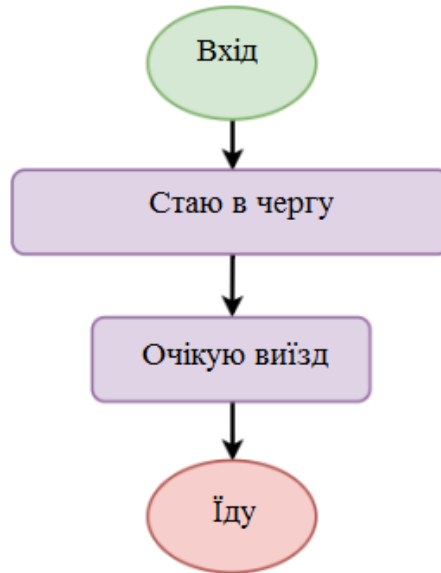


Рисунок 2.11 – Поведінка водія на фазі 3

Час обслуговування при в'їзді, стоянці та виїзді. На всіх трьох фазах є пристрої для обслуговування. На фазі 1 це проїзд ТЗ на територію паркування через конкретний в'їзд. На фазі 3 це вже проїзд ТЗ з території паркування, за її межі. А на фазі 2 є відразу два різні типи пристроїв - пошук, який моделює їзду ТЗ по території паркування з метою знайти місце для стоянки, а другий - сам процес стоянки ТЗ. Всі ці пристрої обслуговування моделюються однаковим способом - за допомогою методу hold або методу 'passivate', який є його аналогом.

На вхід методом 'hold' подається час у секундах, який насправді і є часом обслуговування. Яким буде цей час, залежить тільки від людини, яка проводить імітаційне моделювання. Наприклад, у розділі 3 скрізь був використаний експоненційний розподіл на всіх пристроях обслуговування.

При виконанні моделювання доступні такі види розподілів:

- Бета-розподіл. Двопараметричне сімейство абсолютно безперервних величин, значення яких обмежені до точним інтервалом (зазвичай  $[0, 1]$ ) [12]. На вхід подаються два значення - альфа і бета, які фігурують як експоненти випадкової величини і згідно з якими обчислюється випадкова величина.

- Константа. Не ставить якогось розподілу, просто повертає одне й теж значення.

– Розподіл Ерланга. Двопараметричне сімейство абсолютно безперервних величин, значення яких не обмежені інтервалом, і  $X$  може набувати значень від 0 до нескінченності [13]. На вхід подаються два параметри коефіцієнт форми і коефіцієнт масштабу, згідно з якими і обчислюється випадкова величина.

– Гамма-розподіл. Є аналогом розподілу Ерланга. Точніше якщо параметр (коефіцієнт форми) набуває цілого значення, то такий гамма-розподіл називається розподілом Ерланга.

– Експоненційний розподіл. Абсолютно безперервний розподіл, що моделює час між двома послідовними зверненнями однієї й тієї ж події. На вихід подається якась середня величина, щодо якої і стоїть розподіл.

– Безперервний рівномірний розподіл. Розподіл випадкової речовинної величини, що приймає значення, при належні інтервалу  $[a, b]$ , що характеризується тим, що щільність ймовірності на цьому інтервалі постійна. На вхід подаються два значення, що позначають межі інтервалу, щодо якого і будується розподіл.

– Нормальний розподіл. Розподіл ймовірностей, який в одновимірному випадку задається функцією щільності ймовірності, що збігається з функцією Гауса. На вхід подаються два значення - математичне очікування та середньоквадратичне відхилення, щодо яких і стоїть розподіл.

– Розподіл Пуасона. Імовірнісний розподіл дискретного типу, моделює випадкову величину, що є число подій, що відбулися за фіксований час, за умови, що дані події відбуваються з деякою фіксованою середньою інтенсивністю і незалежно один від одного. На вхід подається середня інтенсивність, щодо якої будується розподіл.

– Трикутний розподіл. Безперервний розподіл імовірностей з нижньою межею  $a$ , верхньою межею  $b$  і режимом  $c$ , де  $a < b$  і  $a < c < b$  [14]. На вхід подається три значення - нижня межа, верхня межа та медіана, щодо яких і стоїть розподіл.

– Розподіл Вейбулла. Двопараметричне сімейство абсолютно безперервних розподілів. На вхід подається два параметри - коефіцієнт форми і коефіцієнт масштабу, щодо яких і будується розподіл [15].

– Функція розподілу. Функція, що характеризує розподіл випадкової величини або випадкового вектора; ймовірність того, що випадкова величина  $X$  набуде значення, менше або дорівнює  $x$ , де  $x$  - довільне дійсне число. На хід приймає словник, в якому описується розподіл ймовірностей .

- Щільність ймовірності. Один із способів завдання вірять ніс на евклідовому просторі. На вхід подається словник, де описується розподіл ймовірностей.

Таким чином, при визначенні параметрів імітаційної моделі користувач може описати час обслуговування на пристроях будь-якого з доступних йому розподілів. Інакше, він може описати свій власний розподіл, успадкувавши його від класу `_Distribution`.

## **3 РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ**

Упродовж всього часу моделювання відбувається збір статистик із кожного компонента. Допомагає у цьому спеціальний клас Monitor.

Monitor - клас, який збирає дані протягом усього часу симуляції . Він автоматично збирає інформацію з ресурсів та черг. За потреби можна визначити свій клас для моніторингу. Також цей клас дозволяє отримати деякі статистики (такі як мінімум, максимум, середнє тощо) за підсумками симуляції або побудувати текстові гістограми .

Також був розроблений власний клас, що дозволяє збирати статистики з усіх компонентів. Цей клас ініціалізується для кожного об'єкта типу Car, що моделює поведінку ТЗ та зберігає статистику на кожному етапі.

### **3.1 Параметри модельованого КВ**

Для більш гнучкого моделювання було визначено низку параметрів, з яких можна налаштувати модель. Список доступних для редагування параметрів наведено на таблиці 3.1.

Дані параметри визначаються у вигляді змінних, які використовуються в компонентах. На основі цих параметрів було отримано результати, що розглядаються далі.

Для завдання часу обслуговування на пристроях використовувався експоненційний розподіл.

Таблиця 3.1 – Список доступних параметрів для моделі

Фаза	Назва	Значення, яке використовується	Опис
Для всього додатка	_loops	90 (днів)	Кількість циклів моделювання (кількість днів)
Вхідний потік	till	86399 (секунд)	Тривалість моделювання в межах одного циклу (день у вигляді секунд)
Вхідний потік	cars_per_day	5000 (Авт./День)	Середня кількість Авт за один цикл (інтенсивність Авт)
В'їзд	number of inputs	2	Кількість в'їзді
В'їзд	input queue size	2	Розмір черги на в'їзді
В'їзд	mean_input_time	20 (секунд)	Середній час обслуговування при в'їзді
В'їзд	max_shared_queue_length	10	Розмір загальної черги на в'їзді, при якому Авт відмовляється від в'їзду
В'їзд	mean_shared_queue_timeout	60 (секунд)	Середній час очікування у спільній черзі, після чого Авт може залишити чергу
Виїзд	max_car_position	4	Якщо після закінчення таймауту, позиція Авт в загальній черзі більша за параметр, то Авт залишає чергу
Стоянка	parking_size	100	Кількість місць для стоянки
Стоянка	mean_search_time	20 (секунд)	Середній час пошуку стоянки
Стоянка	mean parking size	3600 (секунд)	Середній час стоянки
Виїзд	number of outputs	2	Кількість виїздів
Виїзд	output queue size	2	Розмір черги на виїзді
Виїзд	mean_output_time	20 (секунд)	Середній час обслуговування при виїзді

### 3.2 Виведення результатів у вигляді тексту

За замовчуванням Salabim за допомогою класу Monitor надає певні види статистики у вигляді текстових даних. Це - trace лог моделювання, статистики та гістограми.

Трейс-лог показує покрокове виконання моделювання. Таким чином складається враження, що відбувається моделювання в реальному часі. Приклад трейс-логу для одного ТЗ представлений на рис. 3.1.

```

Line 30: 94          car.0 create
Line 31: 94          car.0 activate
Line 45: 99          0.000 car.0 current
Line 46: 108         car.0
Line 47: 109         car.0 hold
Line 64: 164         car.0
Line 65: 169         car.0
Line 70: 191         car.0
Line 448: 205        car.0 activate
Line 448: 109+      3.832 car.0 current
Line 449: 125        car.0 hold
Line 680: 125+      6.106 car.0 current
Line 681: 129        car.0
Line 682: 129        car.0 request honor
Line 683: 129+      6.106 car.0 current
Line 684: 130        car.0 hold
Line 47289: 130+    448.531 car.0 current
Line 47290: 131      car.0
Line 47291: 138      car.0
Line 47292: 139      car.0 passivate
Line 47302: 164      car.0
Line 47303: 169      car.0
Line 47308: 191      car.0
Line 52038: 205      car.0 activate
Line 52040: 139+    498.454 car.0 current
Line 52041: 142+    car.0 ended

```

Рисунок 3.1 – Trace -лог моделювання

У цьому трейс-логі відображається:

- номер рядка в коді, звідки було вчинено дію;
- час (від початку моделювання) коли було скоєно дію;
- об'єкт, який вчинив дію;
- опис того, яку дію було скоєно.

Простеживши за даним логом можна зрозуміти у якому стані була модель у конкретний момент часу. Також даний лог дозволяє вирахувати ряд статистичних даних, наприклад, час очікування при в'їзді.

Для ряду компонентів, таких, як Queue і Resource, Salabim вже ви зчитуєте ряд статистичних показників. Наприклад, на рис. 3.2 показана статистика для вхідної, загальної черги. У верхній частині малюнка відображається статистика про розмір черги, а в нижній - про час перебування черги.

Для обох типів статистик вираховуються такі показники:

- середнє;
- середньоквадратичне відхилення;
- мінімальне значення;
- медіанне значення;
- максимальне значення;
- значення по 90 та 95 перцентилю.

Statistics of input_shared_queue at 86399		all	excl.zero	zero
Length of input_shared_queue	duration	86399	25388.399	61010.601
	mean	1.568	5.336	
	std.deviation	2.863	2.790	
	minimum	0	1	
	median	0	5	
	90% percentile	7	9	
	95% percentile	8	10	
Length of stay in input_shared_queue	maximum	11	11	
	entries	5413	5413	0
	mean	25.026	25.026	
	std.deviation	32.997	32.997	
	minimum	0.000	0.000	
	median	6.915	6.915	
	90% percentile	75.981	75.981	
95% percentile	94.141	94.141		
	maximum	164.146	164.146	

Рисунок 3.2 – Статистика для вхідної загальної черги

Для ресурсів статистика відрізняється. Тут вираховуються ті самі за показниками, проте розділів зі статистикою тут уже більше. Список всіх розділів представлений нижче:

- кількість запитуваних ресурсів;
- час очікування при запиті ресурсу;
- кількість ресурсів, що скористалися;
- час використання ресурсом;
- місткість ресурсу;
- кількість вільних ресурсів;
- кількість зайнятих ресурсів;
- наповненість.

Щодо вибраного розділу можна побудувати гістограму. Вона відобразить скільки часу і в якому стані був обраний ресурс або черга. Наприклад, на рис. 3.3 показана гістограма для стоянки. На ній відображено скільки паркувальних місць було зайнято і впродовж якого часу.



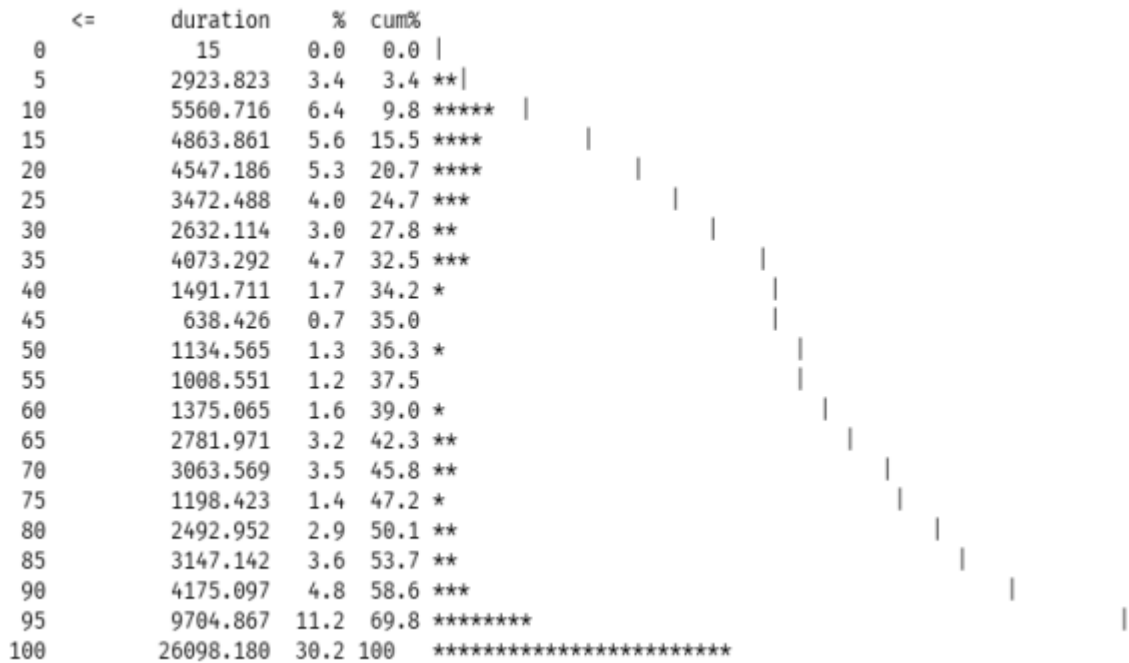


Рисунок 3.3 – Гістограма зайнятих місць на стоянці впродовж одного дня

Крім цього, є ряд розширених параметрів, які зчитуються для різних процесів. Наприклад, кількість ТЗ, що залишили модель у той чи інший момент часу; середній час обслуговування на конкретному в'їзді; кількість ТЗ, що їхали через безпосередній в'їзд. Деякі зі статистик показано на рис. 3.4.

```

Всього за добу проїхало: 5698
З них:
Залишило чергу через велику кількість ТЗ на в'їзді: 107
Залишило чергу через довге очікування в черзі на в'їзді: 595
Залишило чергу, так як не змогли знайти місце: 2563
В'їхало через в'їзд №0: 2818
В'їхало через в'їзд I: 2270
Виїхало через виїзд №0: 2932
Виїхало, через виїзд I: 2156

```

Рисунок 3.4 – Додаткова статистика за підсумками моделювання.

### 3.3 Виведення результатів у вигляді графіків

Для більш наочного представлення результатів моделювання було

розроблено модуль, що дозволяє виводити статистичні дані чи інші результати моделювання як графіків. Для малювання графіків використовується бібліотека matplotlib. Ця бібліотека написана на Python і надає широкий набір API функцій для малювання.

Дані, що зображуються на графіках, збираються за допомогою власного модуля статистики. Користувачеві пропонується вибрати один із типів даних та відобразити його графік.

Нижче, на рис. 3.5 і 3.6, зображені графіки, що показують імовірність того, що на парковці буде зайнято  $N$  паркувальних місць. На рис. 3.5 зображено результати моделювання, коли на парковці 100 місць для стоянки. На рис. 3.6 зображені результати моделювання, коли на паркуванні 200 місць для стоянки. Всі інші параметри цих двох тестів однакові, змінювалася лише кількість місць для паркування, яких доступно на паркуванні. На графіку наведено середні показники за три місяці. Іншими словами, обчислювалася середня величина за 90 циклів повторення.

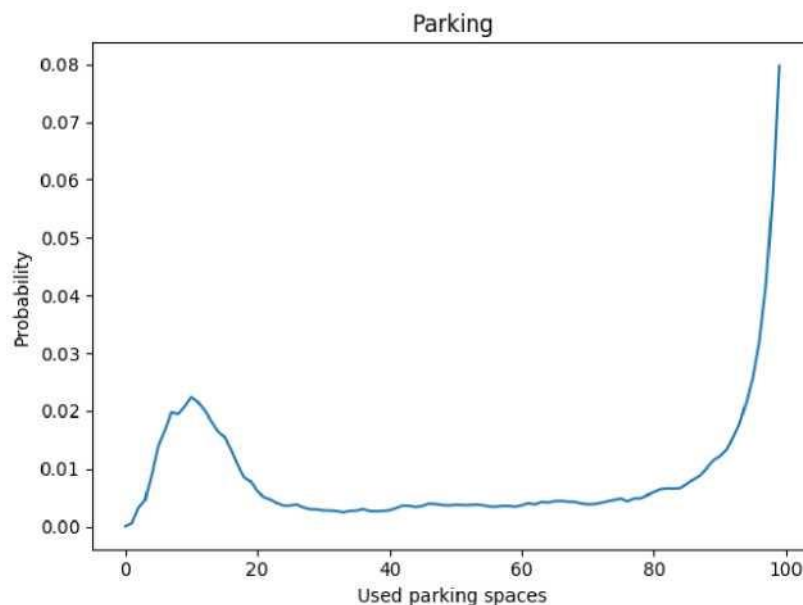


Рисунок 3.5 – Імовірність того, що буде зайнято  $N$  паркувальних місць, якщо кількість місць для паркування рівна 100

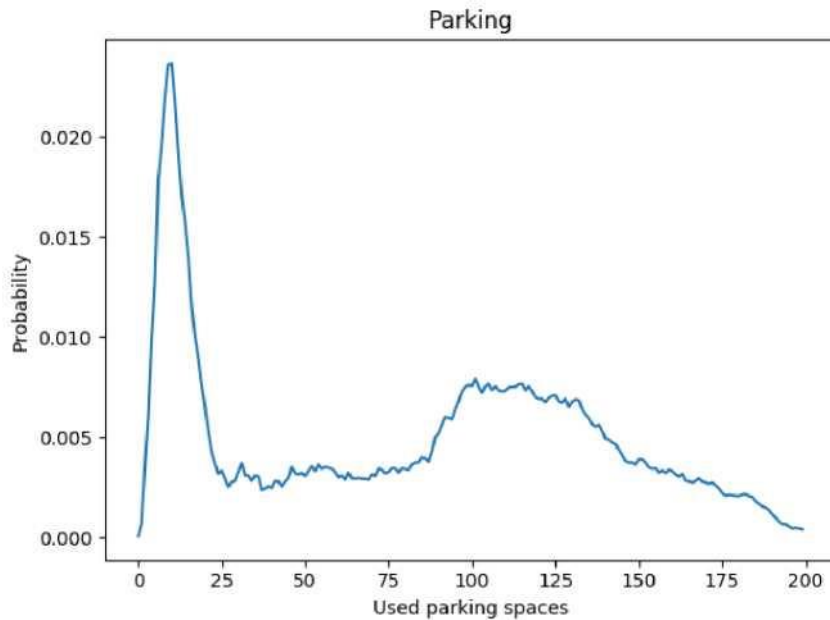


Рисунок 3.6 – Імовірність того, що буде зайнято N паркувальних місць, якщо кількість місць для паркування рівна 200

На рис. 3.7 і 3.8 показано, як зміниться час очікування при в'їзді ТЗ на парковку (від моменту попадання в чергу, до моменту в'їзду на територію стоянки), якщо очікуваний, середній потік Авт збільшиться в 10 разів.

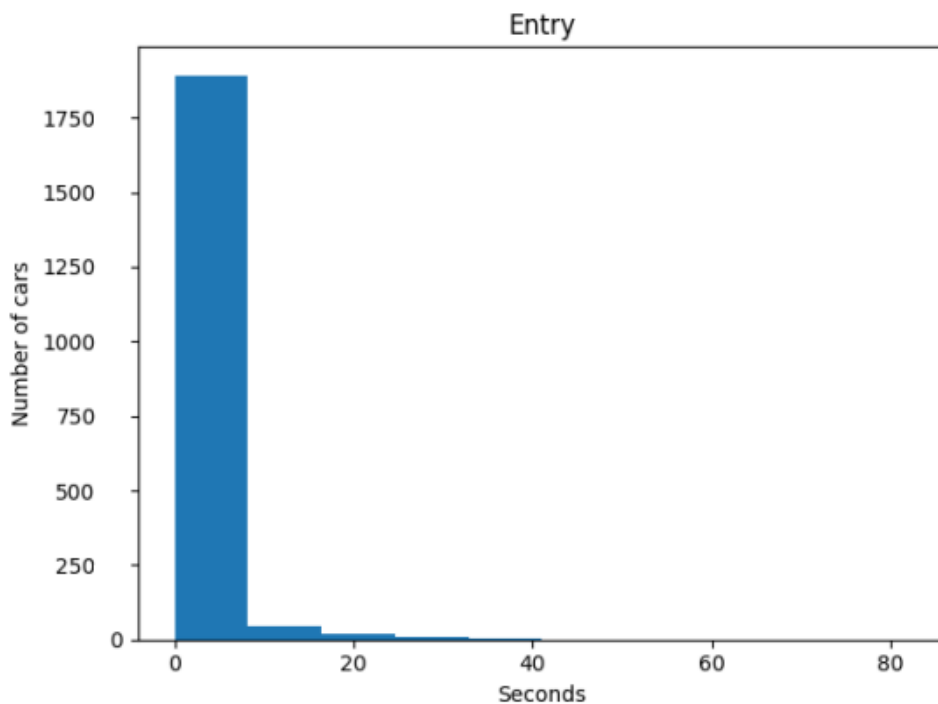


Рисунок 3.7 – Час очікування на в'їзд, за середньої інтенсивності 2000 ТС/день.

На рис. 3.7 показана ситуація, коли очікувана середня інтенсивність дорівнює 2000 ТЗ на день.

На рис. 3.8 показана ситуація, коли очікувана середня інтенсивність дорівнює 20000 ТЗ на день

Аналогічні графіки будуються і для виїзду ТЗ з території паркування, проте вони будуть опущені.

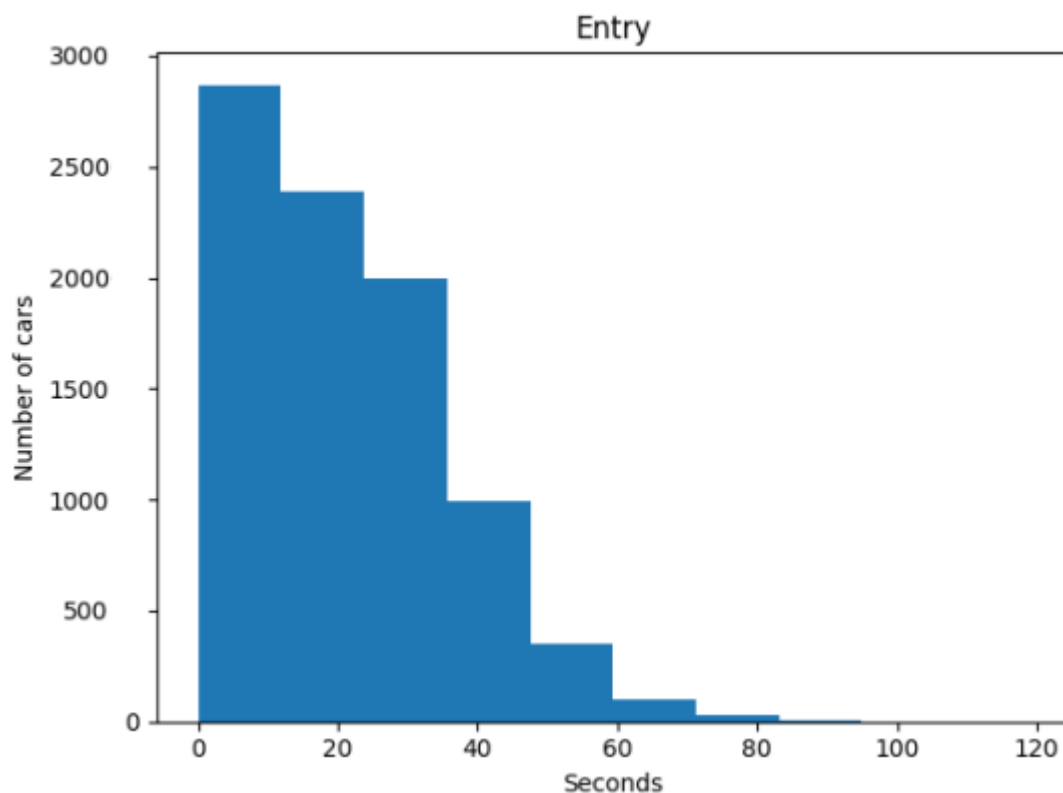


Рисунок 3.8 – Час очікування на в'їзд, за середньої інтенсивності 20000 ТС/день

Як говорилося в попередньому параграфі, є можливість вибрати тип розподілу, згідно з яким генерується час обслуговування на пристроях. За замовчуванням, як і в тестах вище, використовується експоненційний розподіл. Якщо замінити його на пуасонівський, який схожий на експоненційний, і повторити тести, то отримаємо результати, зображені на рисунках 3.9 і 3.10.

На рис. 3.9 зображено графік, що показує ймовірність того, що зайняти N місць, якщо місткість паркування 200 місць.

На рис. 3.10 зображено графік, що показує середній час очікування для ТЗ, що в'їжджає на паркування, за середньої добової інтенсивності 20000 ТЗ на

день.

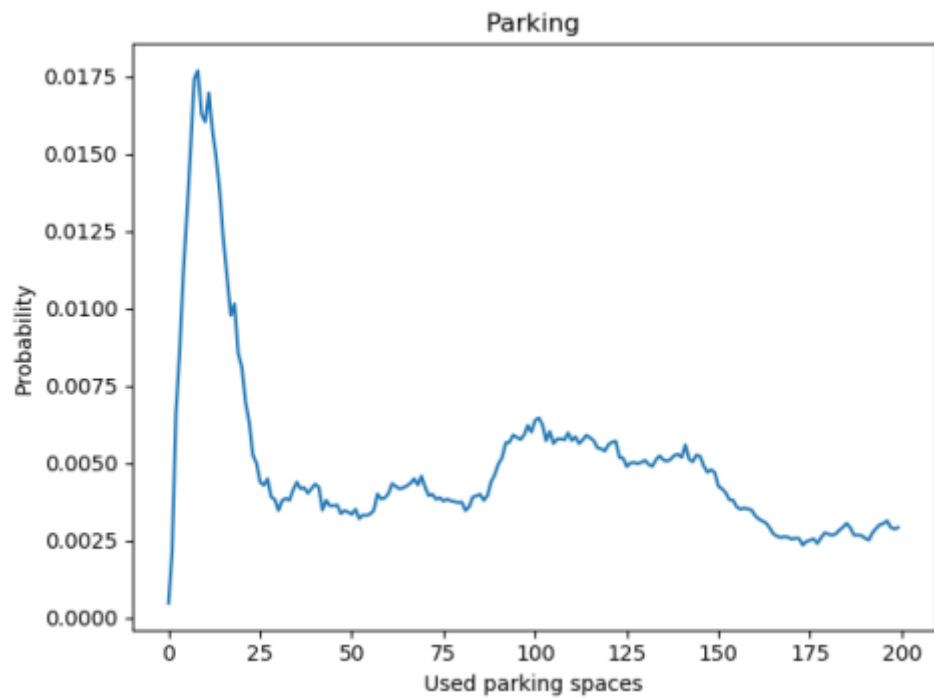


Рисунок 3.9 – Імовірність того, що буде зайнято  $N$  паркувальних місць, якщо кількість місць для паркування рівна 200 (використано розподіл Пуассона)

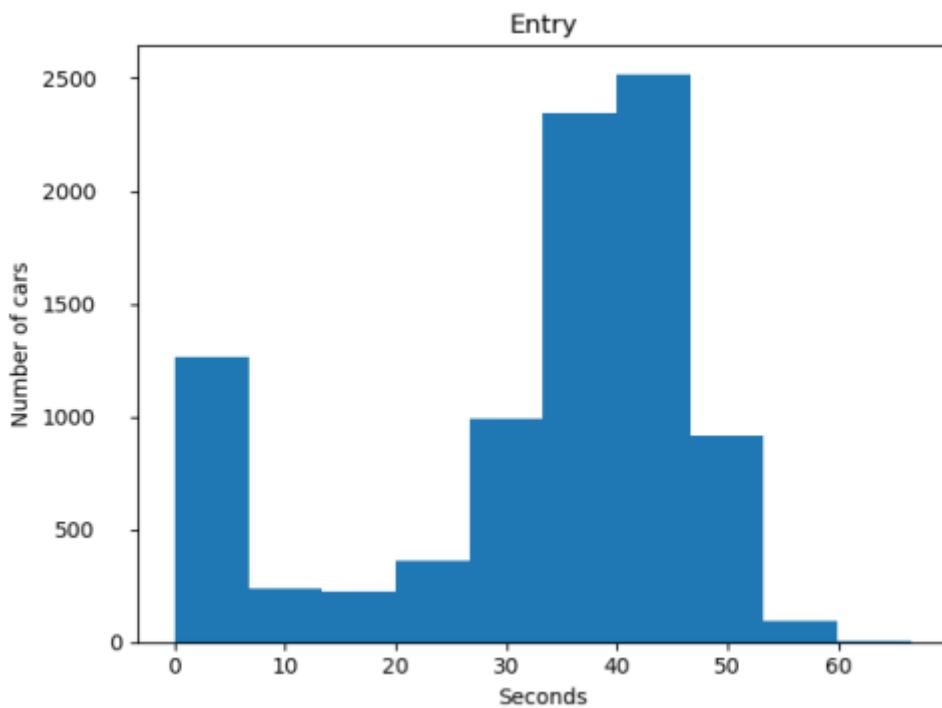


Рисунок 3.10 – Час очікування на в'їзд, за середньої інтенсивності 20000 ТС/день (використано розподіл Пуасона)

## **4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ**

### **4.1 Вимоги ергономіки до організації робочого місця оператора ПК**

Робоче місце — це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників [16].

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпусу, рук, ніг. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла. Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає на 15–25% [17].

Організація робочого місця користувача ПК має відповідати ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги», ДСан ПіН 3.3.2.007-98, характеру та особливостям трудової діяльності.

Площа одного робочого місця користувача ПК повинна складати не менше 6 м<sup>2</sup>, а об'єм — не менше 20 м<sup>3</sup>. Конструкція робочого місця користувача ПК повинна відповідати сучасним вимогам ергономіки, характеру виконуваної роботи і забезпечити оптимальне розміщення на робочій поверхні документів та обладнання ПК (монітора, системного блоку, клавіатури, мишки та інших периферійних пристроїв. Монітор на робочому місці встановлюється так, щоб верхній край екрана знаходився на рівні очей.

Розташування монітора ПК має забезпечувати:

- безпечність роботи в цілому;
- зручність та ефективність зорової роботи з екраном в вертикальній площині під кутом 300 від лінії зору, площа екрана при цьому має бути

перпендикулярною нормальній лінії зору користувача.

Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300мм від краю, ближчого до користувача. Кут нахилу клавіатури має бути в межах 5-15°. Поверхня клавіатури повинна бути матовою з коефіцієнтом відбиття 0,4. Клавiші клавіатури мають бути зручними в роботі і м'якими при натисканні (хід всіх клавiш має бути однаковим з мінімальним опором натискання 0,25Н та максимальним – не більше 1,5Н) [18].

При розміщенні робочих місць з ПК слід дотримуватися вимог, зазначених в ДНАОП 0.00-1.31-99:

- робочі місця розміщуються на відстані не менше 1м від стін з світловими прорізами;
- відстань між бічними поверхнями моніторів ПК має бути не менше 1,2м;
- відстань між тильною поверхнею монітора одного ПК та екраном монітора іншого ПК має бути не меншою 2,5м.

Вимоги двох останніх пунктів враховуються також при розміщенні робочих місць з ПК в суміжних приміщеннях з урахуванням конструктивних особливостей стін та перегородок.

Загальні принципи організації робочого місця:

- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;
- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;
- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою – справа;
- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплення його двома руками;
- робоче місце не повинно бути захаращене;
- організація робочого місця повинна забезпечувати необхідну оглядовість.

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза – це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпусу, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук – запалення суглобів.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам (рисунок 4.1).

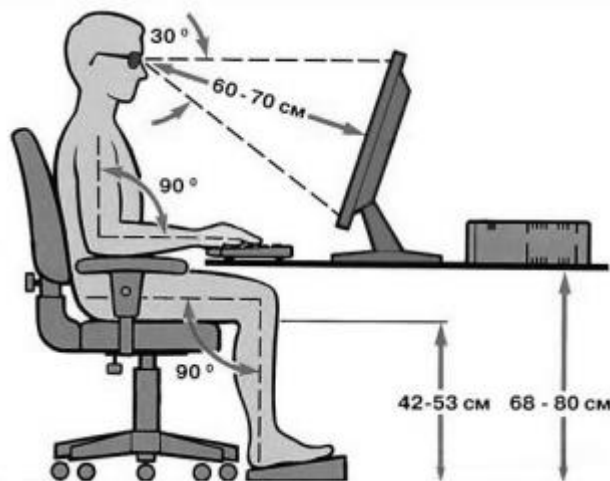


Рисунок 4.1 – Робоче місце і робоча поза користувача ПК

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях – сидячи.

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10% більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта [18].

#### **4.2 Заходи захисту від випромінювань оптичного діапазону**

До випромінювання оптичного діапазону відносяться інфрачервоні й ультрафіолетові хвилі, видиме світло, лазерне випромінювання.

По фізичній природі інфрачервоні промені мають хвильові (довжина хвилі



0,78-540 мкм) і квантові властивості. Генератором випромінювання є будь-яке тіло, температура якого вище абсолютного нуля. За законом Стефана-Больцмана інтегральна густина випромінювання, Вт/м<sup>2</sup>, абсолютно чорного тіла пропорційна четвертому ступеню його абсолютної температури [16].

З підвищенням температури тіла змінюється спектральний склад його випромінювання. Чим вища температура тіла, тим коротша довжина хвилі, максимального випромінювання.

Інфрачервона енергія, яка потрапляє на тіло людини, діє передусім на незахищені його частини (лице, руки, шию, груди), причому конвективне тепло впливає на зовнішній шкіряний покрив, тоді як інфрачервоне випромінювання може проникнути на деяку глибину в тканину. При довготривалому перебуванні людини в зоні інфрачервоного випромінювання, як і при систематичній високій температурі настає різке порушення теплового балансу в організмі. Для вимірювання густини потоку випромінювання на робочому місці застосовують актинометр – прилад, який дозволяє вимірювати густину потоку інфрачервоного випромінювання у діапазоні від 0 до 14кВт/м<sup>2</sup>. Основні види захисту від інфрачервоного випромінювання – захист часом, захист віддалю, усунення джерела тепловиділення, теплоізоляція, охолодження гарячої поверхні, забезпечення тепловіддачі тіла людини та індивідуальні засоби захисту. Потужність випромінювання можна знизити за рахунок конструкторських і технологічних рішень (змінюючи нагрівання виробів у нагрівальних пічках індукційним нагріванням та ін.) і за рахунок покриття поверхні, яка нагрівається, теплоізолювальним матеріалом. Для захисту очей застосовують світлофільтри зі спеціального жовто-зеленого або синього скла.

Ультрафіолетове випромінювання змінює склад виробничої атмосфери. Утворюється озон, оксиди азоту і пероксид водню. Короткохвильове випромінювання іонізує повітря, утворює в атмосфері ядра конденсації, які зменшують освітленість робочих місць і призводять до утворення туманів.

Основні засоби захисту. Першочергові заходи – це конструкторські і технологічні рішення, які виключають генерацію або знижують інтенсивність випромінювання. Спеціальні засоби захисту (екранування джерел

випромінювання, фарбування стін у світлі кольори) попереджують розповсюдження і знижують інтенсивність цих випромінювань у виробничих приміщеннях. Очі захищають окулярами або щитками зі склом – світлофільтром. Для захисту шкіри використовують мазі з речовинами – світлофільтрами для цих променів (салол, саліцилово-метиловий ефір та ін.), а також спецодяг з бавовняних тканин і грубововняного сукна. Руки захищають рукавицями [17].

Діапазон довжин хвиль які випромінюють оптичні квантові генератори (ОКГ) – лазери, охоплює видимий спектр і розповсюджується в інфрачервоній і ультрафіолетовій областях.

Найбільш чутливими до дії випромінювання ОКГ є очі. Випромінювання викликають опіки і пошкодження сітківки ока, це може призвести до сліпоті. Небезпечне не тільки пряме випромінювання, але й відбите від стін, обладнання.

Існують спеціальні норми, до яких ввійшли організаційні та інженерно-технічні заходи, які можуть забезпечити зменшення густини потоків енергії (потужності) на робочих місцях до величин, значно менших від допустимих. ОКГ розміщують в окремих або відгороджених приміщеннях. Саме приміщення і обладнання не повинні мати дзеркальної поверхні. Стіни, стелі, обладнання й інші предмети фарбують матовою фарбою з малою сорбційною здатністю. Приміщення повинно мати високу освітленість, а також припливно-витяжну вентиляцію. При розміщенні в одному приміщенні декількох ОКГ їх огорожують ширмами, шторами або екранами, що не пропускають випромінювання. Надійним захистом від випадкового попадання випромінювання на людину є світловод, який екранує промінь на усьому шляху його дії (від ОКГ до мішені) [17].

## ВИСНОВКИ

У рамках виконання кваліфікаційної роботи було досягнуто наступних результатів:

- розглянуті, вивчені та побудовані стохастичні моделі КВ ТП. Була запропонована модель , що є багатофазною СМО, що дозволяє змоделювати поведінку КВ;

- розроблено метод моделювання ТП, що входить на КВ. Метод створює нестационарний пуассонівський потік, що надходить на вхід розробленої моделі;

- реалізовано додаток, що дозволяє моделювати поведінку КВ, а також проводити імітаційне моделювання КВ. Програма використовує python фреймворк Salabim і надає результати в текстовому та графічному вигляді;

- проведено імітаційне моделювання КВ, взявши за основу паркування.

У рамках моделювання було проведено порівняння моделей з різними параметрами.

Таким чином всі поставлені задачі були вирішені, а цілі досягнуті.

Як плани на подальшу розробку даної моделі поставлено цілі, пов'язані з поліпшенням точності моделювання, порівняння імітаційної моделі з аналогами її реального світу та доопрацюванням додатка, додавши можливість проводити моделювання в реальному часі, відображаючи процес моделювання у вигляді анімації.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кавун-Мошковська Ольга, Мазаракі Анатолій, Ільченко Наталя. Проектування торговельних об'єктів. – К. : КНТЕУ, 2018. – 360 с.
2. Лобашов О.О. Моделювання впливу мережі паркування на транспортні потоки в містах: монографія. –Х.: ХНАМГ, 2010. – 170 с.
3. Лабскер Л.Г. Вероятностное моделирование в финансово-экономической области – М.: Альпина Пабlishер, 2002. – 224 с.
4. Одарченко Микола, Василець Катерина, Тихонченко Руслан. Проектування магазинів та торговельних центрів. – Х. : ХДУХТ, 2017. – 180 с.
5. Salabim website [Електронний ресурс]. – Режим доступу: <http://www.salabim.org/manual/> (дата звертання: 10.04.2022).
6. Стельмах О.В. Містобудівні принципи й методи формування систем паркування легкових автомобілів у великих та найбільших містах України (на прикладі міста Києва): автореф. дис... канд. техн. наук: 05.23.20 / Київ. 2004. 16 с.
7. Сильянов В.В. Теория транспортных потоков в проектировании дорог и организации движения. – М.: Транспорт, 1977. – 303 с.
8. Семенов В.В. Математическое моделирование динамики транспортных потоков мегаполиса. – М., 2004. – 44с.
9. Моделирование пуассоновского процесса [Електронний ресурс]. – 2014. – Режим доступу: <https://habrahabr.ru/post/210432/> (дата звертання: 01.02.2022).
10. Давідіч Ю. О. Конспект лекцій з дисципліни «Моделювання транспортних систем» (для магістрів усіх форм навчання спеціальності 275 – Транспортні технології) / Ю. О. Давідіч, Г. І. Фалецька; Харків. нац. ун-т. міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 71 с.
11. SimPy website [Електронний ресурс]. – Режим доступу: <http://www.simpy.readthedocs.io/en/latest> (дата звертання: 10.04.2022).
12. Beta distribution [Електронний ресурс] – Режим доступу: [http://ru.wikipedia.org/wiki/Beta\\_distribution](http://ru.wikipedia.org/wiki/Beta_distribution) (дата звертання: 20.05.2022).
13. Erlang distribution [Електронний ресурс] – Режим доступу:

[http://ru.wikipedia.org/wiki/Erlang\\_distribution](http://ru.wikipedia.org/wiki/Erlang_distribution) (дата звертання: 20.05.2022).

14. Triangular distribution [Електронний ресурс] – Режим доступу: [http://ru.wikipedia.org/wiki/Triangular\\_distribution](http://ru.wikipedia.org/wiki/Triangular_distribution) (дата звертання: 20.05.2022).

15. Weibull distribution [Електронний ресурс] – Режим доступу: [http://ru.wikipedia.org/wiki/Weibull\\_distribution](http://ru.wikipedia.org/wiki/Weibull_distribution) (дата звертання: 20.05.2022).

16. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. – 267 с.

17. Яремко З. М. Безпека життєдіяльності: Навч. посіб. — Львів., 2005. – 301 с.

18. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. – К.; Каравела, 2004. -328 с.